Selección

commodore Magazine

AÑO I - Núm. 9 - Noviembre 1984 - 250 Ptas.

REVISTA INDEPENDIENTE PARA USUARIOS

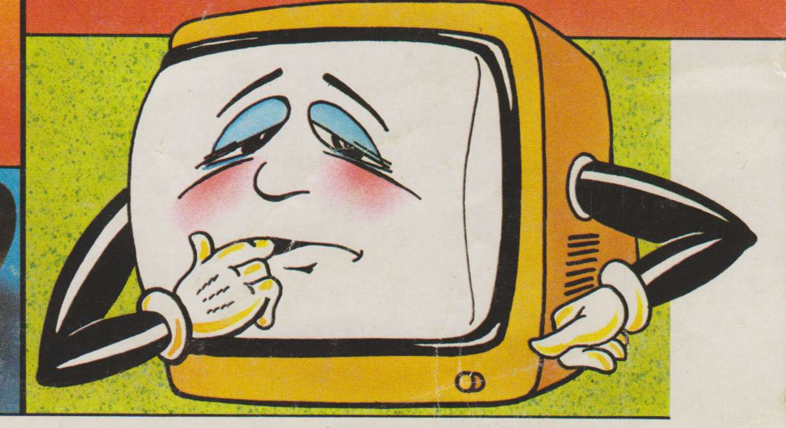


Programas para Vic-20



Móntate un paddle

Identifica tus errores



commodore Magazine

Sumario

DIRECTOR:
Alejandor Diges
COORDINADOR EDITORIAL:
J. A. Sanz
REDACCION:
Roberto Menéndez
Anibal Pardo
Teresa Aranda
Gumersindo García
Fernando García
DISEÑO:
Ricardo Segura

Editada por
PUBLINFORMATICA S.A.
PRESIDENTE:
Fernando Bolín
DIRECTOR EDITORIAL
Norberto Gallego

Administración
INFODIS S.A.
CONSEJERO DELEGADO:
Fernando Bolín
GERENTE DE CIRCULACION Y VENTAS
Luis Carrero
PRODUCCION
Miguel Onieva
SERVICIO CLIENTES
Antonio Zurdo
JEFE DE PUBLICIDAD
María José Martín

Dirección:
Redacción y administración
C/Bravo Murillo, 377.
28020 Madrid
Tel. 733 74 13

Publicidad Madrid:
C/ Bravo Murillo, 377.
28020 Madrid
Tel. 733 96 62/96

Publicidad Barcelona:
Roberto Rodríguez
C/Pelayo, 12. 08001 Barcelona
Tel. (93) 301 47 00 ext. 27

Depósito Legal: M-6622-1984
Distribuye: S.E.G.L.
Avda. Valdelaparra, s/n.
Alcobendas. Madrid

Fotocomposición: Consulgraf
C/Nicolás Morales, 34. Madrid
Fotomecánica: Karmat
C/Pantoja, 10. Madrid
Imprime: Novograph, S.A.
Carretera de Irún, Km. 12,450
Madrid.
Solicitado control O.J.D.

Año 1
Núm. 9

SUSCRIPCIONES

Rogamos dirija toda la correspondencia relacionada con suscripciones o números atrasados a:
COMMODORE
MAGAZINE
EDISA
LOPEZ DE HOYOS, 141
5.º
28002-MADRID

Editorial

Este número está parcialmente dedicado a los poseedores del **Vic-20.** Por algunas de vuestras cartas se deja ver una idea, pensáis que no nos acordamos de vosotros y esto no es así. Ocurre que los programas son algo que no se improvisa y hay momentos en que los disponibles no están a la altura del nivel que pretendemos implantar en la revista, pero también ocurre que cuando llegan tratamos de incluir el máximo número de ellos que nos es posible en tan pocas páginas. De cualquier manera, ya tenemos regularizado un *stock* de buenos programas y artículos tanto para el **Vic-20** como el **C-64.** No obstante queremos recibir vuestras colaboraciones, estamos seguros de que serán buenas. A más de un comodorero le ha llegado la sorpresa (en forma de cheque por colaboración), después de enviarnos el resultado de su trabajo. Tampoco creáis que la revista sólo acoge colaboraciones de programas exclusivamente de juegos. Las experiencias que consideréis útiles e interesantes para los demás tienen igualmente su espacio garantizado. Es de suponer que no utilizáis el ordenador solamente para jugar a los marcianitos. Tal vez lo empleáis como ayuda en vuestros estudios. Esas experiencias pueden ser importantes para otros usuarios, no os las guardéis.

Por otro lado, hemos incluido un artículo, que da comienzo a otros que vendrán después sobre temas más específicos, y nos ofrece una serie de pautas para intentar traducir la mayoría de programas que teníamos en el **Vic-2Ø,** cuando decidimos comprar un **64.** Muchas de las ideas sirven para desarrollarlas en sentido contrario.

Otro artículo que nos pareció interesante es el que explica a qué son debidos los mensajes de error. En realidad es una ampliación de lo que dice el manual. Lo importante es detectar e identificar el error para corregirlo. Os lo recomendamos especialmente a los que nos hacéis saber que sale un mensaje ?OUT OF DATA ERROR, cuando transcribís un programa publicado en **Commodore Magazine,** que lleva muchas sentencias DATA.

Por último, sólo nos queda desearos que disfrutéis del *paddle* que con toda seguridad os fabricaréis, después de leer el artículo correspondiente.



P.: En primer lugar el felicitarnos por vuestra revista, es de una gran ayuda para mí y creo que para muchos.

Otro de los motivos de mi carta es el pediros la dirección de la distribuidora de la cinta **Tales of the Arabian Nights, ABC Soft,** me gustaría comprarles ese juego, pero no sé dónde dirigirme.

Os envío también la encuesta, estoy seguro de que con ella podréis conocer mucho mejor nuestros gustos y preferencias.

S. Meroño
Murcia

R.: Gracias por tus elogios a nuestra revista (de todos los comodoreros).

Esta cinta, cuyo título respondería en castellano liso a **Las Mil y una Noche** (sugestivo nombre, eh?), pertenece a la firma británica **Interceptor Micro's** y fue una de las grandes atracciones de la pasada Feria **Commodore,** celebrada el pasado junio en Londres. Como nos indicas, en nuestro país la está distribuyendo la firma ABC Soft y su dirección es:

ABC Analog. Santa Cruz de Marcenado, 31. Despacho 3-14. Madrid-8. Tel. (91) 248 82 13.

Aprovechando tu carta, damos las gracias a todos los que nos estáis enviando la respuesta a la encuesta y que por tanto participaréis en el próximo sorteo de las 500 cintas de juegos a celebrar a finales de octubre.

P.: Resido en un pueblo de la zona norte-centro de la provincia de Valencia.

Como podéis imaginar, a nadie le hace gracia que esté dos horas trabajando con un programa en el ordenador y en ese momento se vaya la corriente eléctrica y te quedes con dos palmos de narices.

Bueno, con respecto a eso, os comunico que eso me ocurre muy a menudo, y para colmo la instalación de mi piso favorece a aumento de luz instantáneo, eso hace saltar el automático o se carga los aparatos eléctricos.

Por eso me gustaría que me dijeráis si puedo poner un acumulador de energía para que me dé tiempo a guardar el programa (30 seg.), o el paso automático al funcionamiento en pilas.

J. Carrión
Requema. Valencia

R.: Nos hacemos cargo de tu indignación con los cortes de fluido eléctrico. Por otra parte la falta de estabilidad de tensión, que propicia el deterioro de los dispositivos alimentados desde la red, es otro pesado inconveniente, aunque más sencillo de solventar con la ayuda de fusibles (el **Commodore 64** dispone de uno en su interior).

Para solucionar el otro problema, existen a grandes rasgos dos soluciones posibles. La primera y más costosa consistiría en que compraras un dispositivo llamado inversor, que convierte la tensión de una batería de 12 voltios (de coche) en 220 voltios/corriente alterna. También se haría necesario un cargador de baterías.

La solución que tú apuntas, utilizar un acumulador o baterías es más compleja, si no deseamos modificar la circuitería del ordenador. Te explicamos el proceso que utiliza el **C-64** para alimentarse. El módulo de menor tamaño contiene un transformador que convierte la tensión de 220 v. en nueve voltios en corriente alterna, posteriormente pasa a un puente rectificador, que seguido por un estabilizador 7805 proporciona una tensión de cinco voltios estabilizados. Simultáneamente estos nueve v/c.a. pasan a un circuito duplicador de tensión y a partir de él, con ayuda de un 7812, se obtienen doce voltios estabilizados, y otra tensión rectificada en media onda, no estabilizada, de nueve voltios. Por si esto fuera poco, a partir de los nueve v.c.a. procedentes del transformador, se obtiene una señal de 50 ciclos por segundo que ataca a la segunda **CIA 6526** del ordenador. Si bien las tensiones continuas de +5 y +12 voltios se conseguirían fácilmente con baterías, las tensiones alternas requieren circuitería algo más compleja.

La solución más aconsejable es algo más incómoda y económica. Consiste en utilizar la unidad de *cassettes,* grabando el programa a medida que vas avanzando con él, de tal manera que si se produce un corte de energía, solamente habrás perdido unas pocas líneas.

P.: Tengo un **Commodore 64** y una impresora **Seikosha GP 100 A Mark II** y no consigo hacerlos funcionar juntos. Agradecería cualquier tipo de información para conseguirlo. ¿Necesito el *interface* RS-232? Si es así, que programa necesitaría. ¿Se puede conectar directamente al *port* de usuario?; ¿qué programa necesito? En la "Guía de Referencia del Programa del **C-64"** no hay quien se aclare.

F. Ramajo
Albacete

R.: No queremos desilusionarte, pero debemos aclarar que has elegido la impresora correcta pero no el modelo. Existe una versión de la GP 100, el Modelo VC que es directamente conectable al *port* de impresora de **Vic-20** y **C-64,** entre otros, sin requerir ningún tipo de modificación.

El **C-64** se basa en estándar RS-232 para comunicarse con los periféricos, pero la mayor diferencia estriba en los niveles de tensión elegidos, que no son los de +12 y —12 voltios, sino otros más bajos. Por eso no se puede hablar propiamente del estándar RS-232 con el **C-64,** pues establece cómo han de ser las señales eléctricas y sus magnitudes para que los dispositivos se pongan de acuerdo durante la transferencia de los datos. Nuestro consejo es que hables con la firma **Dirac,** distribuidora de la marca en España. Su número de teléfono es: (96) 372 88 89. Probablemente te aportarán una solución.

En cuanto a tu observación estamos casi de acuerdo: en la Guía de Referencia no se entiende casi nada de algunas cosas. Deberías ver el original escrito en inglés, no distribuido aquí masivamente.

Cartas

P.: Somos dos hermanos que tenemos hace poco un **C-64**, y nuestra primera preocupación fue ver lo mucho que se calentaba el ordenador y la unidad de alimentación, después de un continuado uso. ¿Es esto malo?; ¿puede causar daño al ordenador?, ¿cuánto puede estar encendido el ordenador sin problemas?.

La segunda petición es más complicada, nos gustaría que pudiésen publicar alguna explicación o artículo con programas de ejemplo (como han hecho con «Gráficos con PEEK y POKE») sobre los registros del *chip* VIC-II.

R.: La preocupación que comentáis es normal. La unidad de alimentación se calienta porque tiene en su interior un transformador, reductor de la tensión de la red. Es sabido que los transformadores se calientan debido a unas corrientes de pérdida, llamadas de Focault, que se genera en el núcleo metálico como efecto secundario del paso de la corriente eléctrica por los bobinados. Los fabricantes tratan de reducirlos, utilizando diversas técnicas, pero la más corriente y barata es construir el núcleo en base a láminas metálicas en lugar de una pieza maciza.

El ordenador también se calienta principalmente por la zona en que se halla situada la circuitería de alimentación, siendo los circuitos integrados encargados de estabilizar el nivel de la tensión los principales causantes de la emisión de calor, disipado durante su funcionamiento.

Ninguno de estos dos factores es realmente preocupante en sí. El fabricante del ordenador ya los tuvo en cuenta durante la etapa de diseño del aparato. De todas maneras, es una regla de general conocimiento que el calor es el principal enemigo de la vida de los componentes. Cuanto mayor sea el calor más probabilidades existen de acortar la vida de los mismos. El tiempo de utilización queda sometido a la discreción del usuario, pero desde luego no hay establecido límite alguno.

En cuanto a la petición, aclaramos desde aquí que el tema de los registros que incorpora el VIC-II es uno de los muchos artículos que estamos elaborando de cara a futuros números de la revista.

PROGRAMA:
CONTABILIDAD DEL HOGAR
TIPO: GESTION
DISTRIBUIDOR: ABC SOFT
FORMATO: CASSETTE VIC 20 CON AMPLIACION DE BK

El programa que comentamos en esta ocasión está diseñado para controlar los gastos de una casa y que podamos saber dónde se nos va el dinero. Una vez cargado de cinta podemos realizar las siguientes operaciones: introducción de datos estimados. En esta opción introducimos los gastos que creemos que se van a producir durante el mes (hasta 20 distintos)

MENU PRINCIPAL
1. GASTOS ESTIMADOS
2. GASTOS ACTUALES
3. CUENTAS BANCO
4. GRAFICO GASTOS
5. VER GASTOS
6. SALVAR FICHERO
7. CAMBIAR CABECERAS
8. FIN DE PROGRAMA
QUE OPCION?

y que luego se pueden comparar con los que se han producido realmente, que son otros veinte, viendo un gráfico anual de cada gasto o comparándolos todos y viendo la diferencia entre los reales y los estimados. Cada uno de estos gastos tiene un título que se puede cambiar rápidamente mediante otra opción del programa.

Otra rutina que incorpora nos permite llevar el control de la cuenta bancaria, anotando todos los ingresos y gastos que en ella se produzcan durante el año. Posteriormente se le puede pedir que nos muestre él las transacciones realizadas y el saldo actual. Asimismo se puede realizar un cierre periódico que nos borra todos los movimientos de la cuenta, dejando uno único con el saldo anterior.

Las últimas opciones del programa consisten en la grabación y lectura de los datos en cinta, ya que al apagar la máquina se borran de la memoria.

PUNTUACION:
UTILIDAD: 6
PRESENTACION: 5
CLARIDAD: 6
RAPIDEZ: 7

PROGRAMA:
ARROW 64
TIPO: UTILIDADES
DISTRIBUIDOR: CENTRO COMMODORE
FORMATO: CARTUCHO COMMODORE 64

Este paquete incluye un conjunto de utilidades muy útiles para todas las personas que se dedican a la programación, en lenguaje máquina o en BASIC

El primer grupo de órdenes se refiere a la cinta, con el cartucho se puede realizar la grabación y carga de programas a muy alta velocidad, asímismo dispone de un comando para verificar ficheros, y otro más importante, que permite mezclar programas en memoria leyéndolos de cinta; este comando es especialmente útil al no disponer el Commodore de ninguna orden similar. Naturalmente el proceso se realiza a alta velocidad.

El conjunto de instrucciones

ARROW 64
=centro commodore=
SOFTWARE PROFESIONAL

de ayuda al BASIC incluye varios comandos útiles que han sido «olvidados» en el BASIC del ordenador, el primero es la instrucción «AUTO» que posee un gran número de ordenadores. Una vez usada esta instrucción el programa nos va escribiendo los números de línea de programa, ahorrándonos esa labor a nosotros, que sólo deberemos añadir las instrucciones en sí, sin poner el número de línea. Cuando se haya introducido una línea y se pulse «*RETURN*» el programa nos escribirá el número de la siguiente, etc., el usuario puede elegir en qué línea se empieza a imprimir el número y el incremento entre líneas. La segunda instrucción del grupo sirve para borrar varias líneas de golpe, sin el cartucho este proceso llega a ser tedioso al tener que ir tecleando los números uno por uno; con este programa, en cambio, basta indicar el número de la primera y de la última y se borrarán todas las comprendidas entre estas dos. Las dos últimas instrucciones de este grupo sirven para localizar y sustituir textos que se encuentren dentro de un programa

El último bloque de utilidades está dedicado a los programadores muy avanzados y sirve para el manejo del lenguaje máquina, comprendiendo un calculador en hexadecimal (si usted no tiene 16 dedos en la mano) y un monitor que le permite ver los contenidos de la memoria, modificarlos, salvarlos en cinta, etcétera.

PUNTACION:
UTILIDAD: 8.
PRESENTACION: 5.
CLARIDAD: 3.
RAPIDEZ: 8.



VIC-20 A

Si antes tenía un **Vic-20** y ahora ha decidido comprarse un **Commodore 64**, inmediatamente surge la duda sobre si tendrá que dar por perdidos todos los programas anteriores. Con este artículo pretendemos ayudarle a convertir sus programas para el **64**, consiguiendo a cambio un mejor conocimiento de ambos ordenadores.

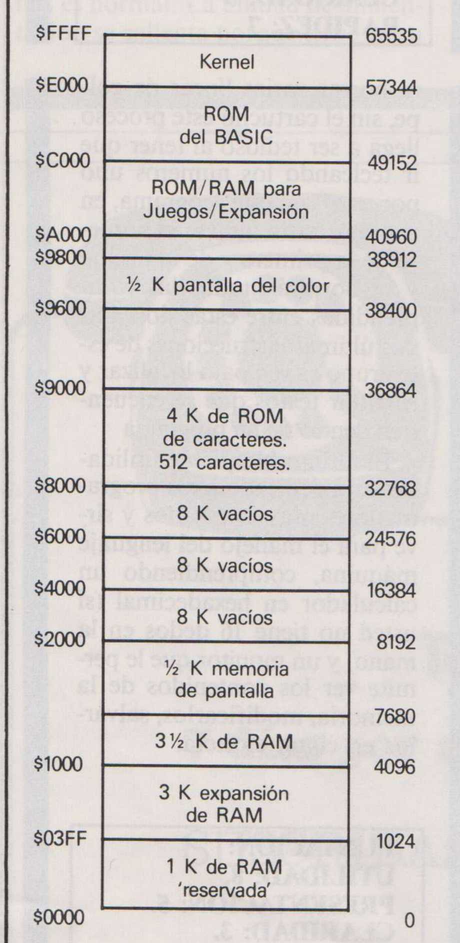
Los programas generalmente consisten en una mezcla, o una, de las siguientes partes:

1. La clásica **palabra clave** del BASIC (ver «El Misterio del BASIC, publicado en el número de agosto), sea REM, PRINT u otra cualquiera.
2. POKES y PEEK que manejan la información de determinadas direcciones de la memoria.
3. Juegos de caracteres definidos por el usuario y gráficos de alta resolución.
4. Efectos sonoros y musicales.
5. Código máquina, sea en la longitud total del programa o formando parte como subrutinas del programa en BASIC.

Si utiliza una unidad de *diskettes* con el **Vic**, entonces no tendrá problemas en cargar los programas directamente en el **64**, pero si utiliza el *cassette*, la cosa se complica. Aunque el **Vic** y el **64** utilizan el mismo *cassette*, los programas del **Vic** guardados en cinta no cargarán en el **64**. Una solución podría consistir en que un amigo nos preste la unidad de *diskettes* para trasvasar los programas de la cinta al disco.

LAS PALABRAS CLAVE DEL BASIC

Tanto el **Vic-20** como el **64** utilizan



Kernel
ROM DEL BASIC
ROM/RAM para Juegos/Expansión
38400 $96000
½ K pantalla de color
37888 $9400
4 K ROM de caracteres
24 K de posible expansión
8192
3½ K
4607
½ K memoria de pantalla
$1000 4096
3 K expansión de RAM
$03FF 1024
1 K de RAM 'reservada'
$0000 0000

Fig. 1: Mapas de memoria del Vic-20.

$FFFF
8 K ROM del Kernel
$E000
4 K ROM de E/S
$D000
4 K RAM
$C000
8 K ROM del BASIC
$A000
8 K RAM/cartucho
$8000
8 K RAM
$6000
8 K RAM
$4000
8 K RAM
$2000
RAM del BASIC
RAM del BASIC
$0800
Pantalla + punteros de Sprites
$0400
1 K de RAM 'reservada'
$0000

Fig. 2: Mapa de memoria del Commodore 64.

COMMODORE 64

el mismo BASIC que puso **Commodore** en su primer ordenador, el **PET**. Cada una de las máquinas dispone de las mismas **palabras clave** del BASIC y se ejecutan de la misma manera exactamente.

En el **Vic** podemos escribir líneas de programa de hasta 88 caracteres de longitud, mientras que en el 64 estamos limitados a 80 (el **64** puede ejecutar líneas de más de 80 caracteres, pero se hacen difíciles de editar).

Tal vez el comando más afectado por los cambios de máquina es la sentencia con PRINT. El **64** dispone de una pantalla de 25 líneas de hasta 40 caracteres, mientras que el **Vic-20** tiene tan sólo 23 líneas de hasta 22 caracteres. Esto significa que usted obligatoriamente debe formatear de nuevo las pantallas realizadas con el **Vic**, pues si no solamente llenaría la mitad izquierda de la pantalla del **64**. El trabajo es bastante sencillo y tedioso, pero se pueden conseguir buenas pantallas alterando los TAB y SPC, no olvidando el mayor tamaño de la pantalla.

Mientras realiza esta tarea podía incluso utilizar los ocho colores extra del **64**. Estos son accedidos utilizando la tecla que lleva el logotipo de **Commodore** y las teclas numéricas.

PEEK Y POKE EN LAS AREAS DE MEMORIA

Los mapas de memoria de las figuras 1 y 2 muestran cuáles son las diferencias entre los mapas de memoria de las dos máquinas. La figura 3 las realza con mayor detalle. Las memorias de color y pantalla de los dos ordenadores trabajan de la misma manera: debemos establecer cuál va a ser el color en la memoria del color, antes de visualizar un carácter en la pantalla. En el **Vic** el contenido de la dirección 36879 dicta cuáles son los colores de la pantalla y el reborde, mientras que en el **64** son manejados de manera separada, con la dirección 53280 para el color del reborde y la 53281 para el de la pantalla. Estas toman un valor de los comprendidos entre 0 y 15, siendo el gris a quien corresponde el 3. Cualquier programa que implique POKE o PEEK con la pantalla, por ejemplo un juego, deberá ser reescrito considerablemente, teniendo en cuenta la mayor pantalla del **64**.

En el **64** cualquier carácter que sobrepase el final de la línea aparece inmediatamente en la primera columna desplazándose durante otros 18 caracteres de la misma línea.

Ordenador	Tamaño de la memoria del BASIC	Posición de la pantalla	Posición de la pantalla del color
Vic sin expansión	4069-7679 ($1000-$1DFF)	7680-8191 ($1E00-$1FFF)	38400-38911 ($9600-$97FF0
VIC + 3 K	1024-7679 ($0400-$1DFF)	7680-8191 ($1E00-$1FFF)	38400-38911 ($9600-$97FF)
VIC + 8 K (o más)	4608-???? ($1200-$????)	4096-4607 ($1000-$11FF)	37888-38399 ($9400-$95FF)
Commodore 64	2048-40959 ($0800-$9FFF)	1024-2023 ($0400-$07E7)	55296-56319 ($D800-$DBFF)

(???.—Los límites superiores dependen del tamaño de la expansión).

Fig. 3: Cuadro de las principales diferencias entre el Vic *y el* Commodore 64.

Del hecho de que el mapa de memoria del **Vic** cambia dependiendo de la cantidad de expansión de memoria utilizada, se derivan más problemas. Un programa para una expansión de 3K tendrá que ser convertido de un modo ligeramente distinto de otro que emplea la de 8 K (la figura 1 nos ayuda a verlo mejor). Los juegos de caracteres definidos por el usuario son manejados de la misma forma en ambos ordenadores. En el **64**, sin embargo, la ROM que almacena el juego de caracteres estandar cohabita con la ROM Entrada/Salida en la dirección 53248 ($D000). Si su programa alberga caracteres propios en forma de sentencias con DATA al final del listado, simplemente tendrá que convertir la parte del programa que efectúa la lectura (READ) de esos datos. La ROM de caracteres del **64** también comienza en la dirección 53248 ($D000), pero antes de poder acceder a ella, el programa que tenemos debe «desconectar» a la ROM de E/S. Esto se lleva a cabo mediante dos sencillos POKE:

POKE 56334, PEEK (56334) AND 254

POKE 1, PEEK (1) AND 251

El juego de caracteres del **64** ya puede ser extraído de la ROM. Igual que con el **Vic**, las letras comienzan con @ en la dirección 53248. Después de que han sido leídos los caracteres

VIC-20 A COMMODORE 64

que requerimos, habrá que recurrir a POKE para que vuelva a ser puesta en su lugar del mapa de memoria la ROM de E/S:

POKE 1, PEEK(1) OR 4

POKE 56334, PEEK(56334) OR 1

Los nuevos juegos de caracteres pueden ser emplazados en el **64**, pero cuando se utilizan con el BASIC, la mejor dirección para ellos es la 14336. Para pasar de un juego de caracteres en ROM a otro situado en la dirección 14336, simplemente hay que POKEar la dirección 53272 con 30. Para proteger nuestro juego de caracteres de no ser alterado por la carga del programa en BASIC, debemos poner el límite del área del programa en la dirección 14080. Esto lo hacemos POKEando las direcciones 52 y 56 con 55, que nos deja libres unos 12 kbytes de memoria para almacenar el programa. En el **Vic** los gráficos de alta resolución se consiguen llenando la mitad de la pantalla con un conjunto de caracteres definidos por el usuario en blanco, después se dobla el tamaño de los caracteres para que los 256 caracteres llenen toda la pantalla. En el **64** las cosas un poco más sencillas, pues disponemos de un área de 8 K de memoria que podemos destinar a la pantalla de alta resolución de 200 por 320 puntos. El modo en alta resolución es accedido como sigue:

POKE 53265, PEEK (53265) OR 32

POKE 53272, PEEK (53272) OR 8

El último de estos POKE dice al **64** dónde hallar la pantalla de alta resolución, en este caso 8192 ($2000). En el **64** los colores para este modo son proporcionados por el contenido de la matriz de la pantalla, situada entre las direcciones 1024 y 2023, mientras que en el **Vic** los colores de los puntos que aparecen dibujados provienen de la pantalla del color. Como en el **Vic**, la pantalla de alta resolución del **64** es depositada como filas de caracteres. Para dibujar un punto (X, Y) en el **64** deberíamos utilizar lo siguiente:

DIR = 8192 + 320 * INT(Y/8) + 8 * INT(X/8) + (Y AND 7)

POKE DIR, PEEK(DIR) OR 2↑(7—(X AND 7))

Esto es similar al método empleado en el **Vic**.

Otro modo que tienen ambos ordenadores en común es el multicolor, que nos permite visualizar hasta cuatro colores en el cuadradito de cada carácter.

En el **Vic** esto es conseguido colocando un código de color mayor que siete en el cuadrado de color correspondiente al carácter requerido. En el **64** se utiliza el mismo método, pero primero debemos activar el modo con la línea siguiente:

POKE 53270, PEEK (53270) OR 16

Después de esto conseguimos que el modo sea prácticamente lo mismo en ambos ordenadores. En los dos la resolución de la pantalla se ha reducido a la mitad, por lo que un par de bits nos da acceso a los cuatro colores, siendo la principal diferencia la dirección desde la que los ordenadores acceden a los colores. La figura 4 muestra las diferencias.

Conformación de los bits	VIC	Alta resolución del Commodore 64	Caracteres del Commodore 64
00	Color de pantalla	Color de la pantalla	Color de la pantalla
01	Color del carácter	4 bits superiores de la memoria de la pantalla	Color de fondo n.º 1 (53282)
10	Color del reborde	4 bits inferiores de la memoria de la pantalla	Color de fondo n.º 2 (53283)
11	Color auxiliar (36878)	Memoria del color	Memoria del color

Fig. 4: Conformación de los bits en el modo multicolor.

EFECTOS SONOROS Y MUSICALES

Una de las principales ventajas que tiene el **64** sobre el **Vic** es el sonido. El **Vic** dispone de tres canales de tono y uno de ruido blanco con una gama de cinco octavas. El **64** tiene tres canales de hasta nueve octavas. Cada uno de los canales del **64** puede disponer de una diferente forma de onda, que puede ser triangular, diente de sierra, impulsos y ruido blanco. El *chip* SID (*Sound Interface Device*) también ofrece una envolvente, pudiendo dar personalidad a la nota, variando los períodos de ataque, decaimiento, sostenimiento y relajación, así como varios filtros. Naturalmente que cuando usted pasa a utilizar el **64**

quiere que el programa suene de la misma forma que lo hacía en el **Vic**. Sin embargo, ahora podrá aumentar la calidad del sonido utilizando las posibilidades del **64**.

CONTROL DE VOLUMEN

Ambos ordenadores manejan el control de volumen de su sonido de la misma forma. Esto lo consiguen POKEando una determinada dirección de memoria con un valor comprendido entre 0 y 15 (siendo 15 el más elevado). En el **Vic**, esta dirección es la 36878 y en el **64** es la 54296.

Cada uno de los tres canales del **Vic** son preajustados para proporcionar la nota dentro de una determinada gama, dependiendo del canal.

En el **64**, cualquier nota dentro de una gama de nueve octavas puede hacerse sonar a través de los tres canales. Hacer que suene una nota en **Vic** es simplemente una cuestión de establecer cuál va a ser el volumen, POKEando después un número comprendido entre 128 y 255 en uno de los tres canales.

Este proceso es ligeramente más complicado en el **64**, en el que el *chip* SID debe saber qué forma de onda bebe utilizar y la envolvente de la nota a ejecutar, antes de poder escuchar algún sonido. Para capacitar que el **64** disponga de una gama mayor que el **Vic**, debemos alterar dos direcciones, para que contengan el tono de cada nota. La sugerencia es que en primer lugar ajuste el registro de Ataque/Decaimiento de la voz que esté utilizando a 136, por ejemplo, y el registro de Sostenimiento/Relegación a cero. También es aconsejable que ponga la forma de onda de la voz en triangular o diente de sierra. Estos valores proporcionan una nota que suena casi igual que las del **Vic 20**. Si el programa del **Vic** utiliza el ruido blanco (dirección 36877) entonces ponga la forma de onda del **64** en ruido.

Una vez que el programa ejecuta adecuadamente, podrá experimentar para obtener mejores efectos de sonido. El Manual del Usuario de **64** proporciona una tabla que indica los valores adecuados para cada nota.

A 10	K 37	U 30	/ 55	6 19	* 49
B 28	L 42	V 231	L 48	7 24	@ 46
C 20	M 36	W 94	+ 40	8 27	↑ 54
D 18	N 39	X 23	– 43	9 32	← 57
E 14	O 38	Y 25	0 35	= 53	SP 60
F 21	P 41	Z 12	1 56	f1 04	
G 26	Q 62	, 47	2 59	f3 05	
H 29	R 17	. 44	3 84	f5 06	
I 33	S 13	; 50	4 11	f7 03	
J 34	T 22	: 45	5 16	RET 01	

Fig. 5: Valor de las direcciones.

CODIGO MAQUINA

La conservación del código máquina del **Vic** al **64** no es tan difícil como pueda parecer a primera vista, si ya dispone de un conocimiento conseguido trabajando con el código máquina. Ambos ordenadores utilizan un microprocesador de la familia 6500. Al igual que ocurre con los BASIC, el código máquina de ambas máquinas es exactamente el mismo, y si lo ha aprendido con el **6502** en el **Vic**, el paso al **6510** le será familiar. Se dará cuenta que las llamadas ilegales a la ROM del **Vic** no funcionan en el **64**, pero las legales sí lo harán, puesto que la tabla de saltos del **Kernel** permanece. El primer 1 Kbyte de memoria es exactamente el mismo en ambas máquinas, excepto para las direcciones 0, 1, 2, 784, 785, 786, 197 y 203 (decimales).

En el **Vic**, las direcciones 1 y 2 guardan la dirección de comienzo de los programas llamados mediante el comando USR, siguiendo la moda tradicional en los **6500**, de byte bajo/byte alto.

En el **64**, las dos direcciones más bajas son utilizadas por el **6510** para el sistema de paginación de la memoria, las llamadas USR en el **64** utilizan las direcciones 784 a 786, por lo que cuando reconvierta programas que utilicen USR recuerde POKEar la dirección de la subrutina en código máquina en las direcciones 785 y 786 del **64**.

Cualquier programa para el **Vic** que emplee las direcciones 784 a 787 para almacenamiento, deberá tener que recurrir a alternativas. Si el programa necesita la dirección 197 ó 203 para saber qué tecla está siendo presionada, debería advertir que el **64** lee el teclado de diferente manera que en el **Vic**. Estas direcciones pasan a contener los valores mostrados en la figura 5.

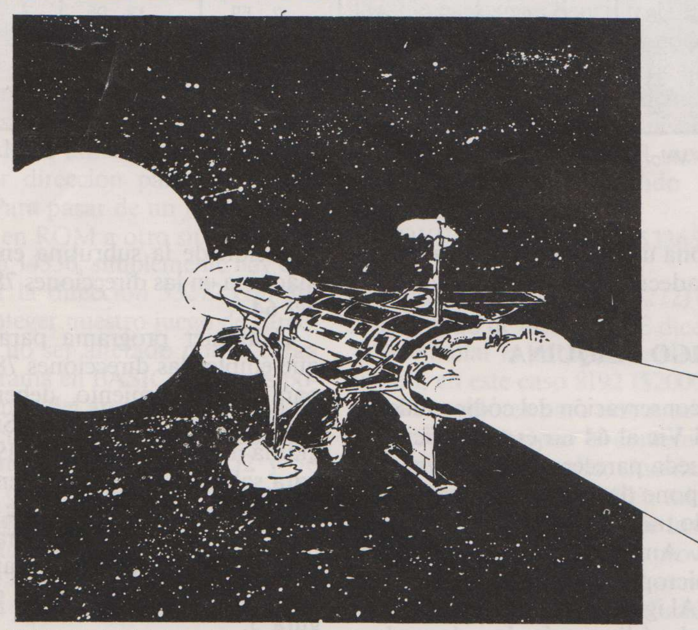
Recuerde que en el **64** dispone de 4 K de memoria separados para recibir programas en código máquina en la dirección $C000 ó 49152). Algunos monitores/ensambladores disponibles para el **Vic** trabajarán con el **64**.

CONCLUSION

Con la información y tablas ofrecidos en este artículo, usted debería ser capaz de convertir la mayoría de los programas de utilidad desarrollados en código máquina, pero los programas de mayor complejidad requieren una mayor cantidad de conocimientos acumulados. No conviene olvidar tampoco que el **64** dispone de otras características tales como los *Sprites*, que necesitan de mucho trabajo en el **Vic 20** para conseguir resultados similares. No obstante, en futuros artículos seguiremos tratando el siempre importante tema de la conversiones de programas entre ambos ordenadores.

Programas

Invasión



Invasión es un clásico entre los juegos de marcianitos y fue uno de los primeros en aparecer en las máquinas de los bares y en las salas de juegos recreativos. Consiste en evitar una invasión que llega de los cielos en forma de varias hileras de marcianitos que van descendiendo inexorablemente. Abajo, y protegido por una serie de bloques defensivos se encuentra el jugador, provisto de un arma láser con la que puede ir destruyendo marcianos. Los marcianitos también disparan, y sus diparos van destruyendo poco a poco las defensas. El juego se controla mediante las teclas "Z" y "C" para desplazarse hacia los lados, mientras que la tecla "M" sirve para disparar. El programa incluye algunas rutinas en lenguaje máquina que se cargan desde las sentencias DATA del final del programa. Por ello es muy importante que dichas sentencias estén correctamente copiadas.

```
10 REM ************************
15 REM *   I N V A S I O N    *
20 REM *                      *
25 REM * COMMODORE  MAGAZINE *
30 REM ************************
35 :
40 V=53248:POKE54276,0:POKE54290,0:POKE54278,240:POKE54292,240:POKE54276,129
45 POKE54290,17:POKEV+33,0:POKEV+32,0:SC=1024:CL=55296:CO=54272:POKE54296,15
50 POKE56,28:POKE55,0:FORI=12544TO12551:POKEI,0:NEXT
55 GOSUB290:GOSUB325:GOTO210
60 IFK=1THEN60
65 IFK<>12ANDK<>20THEN85
70 POKESC+Y*40+X,32:X=X-(K=12)*(X>0)+(K=20)*(X<39):IFPEEK(SC+Y*40+X)<>32THEND=1
75 POKECL+Y*40+X,3:POKESC+Y*40+X,0
80 RETURN
85 IFK=36 AND SX<0THENSX=X:SY=Y-1:POKE54273,72:POKE54272,169
90 RETURN
95 POKESC+SY*40+SX,32:SY=SY-1:IF SY<1THENSX=-1:RETURN
100 P=PEEK(SC+SY*40+SX)
105 POKECL+SY*40+SX,7:POKESC+SY*40+SX,2:IFP=32THENRETURN
110 POKE54273,34:POKE54272,75:POKESC+SY*40+SX,4
115 IFP<>1THEN140
120 FORY1=8TO0STEP-2:FORX1=10TO0STEP-2:P=IN+Y1*6+X1
125 IFPEEK(P)<>SYORPEEK(P+1)<>SXTHEN135
130 S=S+(12-Y1)*5:NO=NO-1:POKEP+1,255:X1=0:Y1=0
135 NEXT:NEXT:GOTO155
140 IFP<>3THEN155
145 FORJ=0TO5:IFS(J)=SC+SY*40+SXTHENS=S+5:S(J)=0:J=5
150 NEXT
```

```
155 POKESC+SY*40+SX,32:SX=-1:RETURN
160 POKES(J),32:S(J)=S(J)+40:IFS(J)>1983THENS(J)=0:RETURN
165 P=PEEK(S(J)):POKES(J)+CO,3:POKES(J),3
170 IFP=32THENRETURN
175 POKES(J),4:POKE54273,43:POKE54272,52:IFP=0THEND=1
180 IFS(J)=SC+SX+SY*40THENSX=-1
185 POKES(J),32:S(J)=0:RETURN
190 S1=INT(RND(1)*6):IFS(S1)>0THENRETURN
195 FORY1=8TO0STEP-2:IFPEEK(IN+Y1*6+S1*2+1)=255THEN205
200 S(S1)=SC+(PEEK(IN+Y1*6+S1*2)+1)*40+PEEK(IN+Y1*6+S1*2+1):Y1=0
205 NEXT:RETURN
210 D=0:FORI=1TOSP:K=PEEK(PE):IFK<>64THENGOSUB60
215 IFSX>=0THENGOSUB95
220 FORJ=0TO5:IFS(J)>0THENGOSUB160
225 NEXT:POKE54273,0:POKE54272,0:POKE54287,0:POKE54286,0
230 IFRND(1)>.5THENGOSUB190
235 NEXT:SYS7168:IFNO<21THENSP=3:IFNO<11THENSP=2:IFNO<6THENSP=1
240 POKE54287,2:POKE54286,37:IFPEEK(12303)=36THENPOKE12303,129:GOTO250
245 POKE12303,36
250 IFPEEK(7604)=0ANDD=0ANDNO<>0THEN210
255 IF NO=0THENGOSUB310:GOTO210
260 POKESC+Y*40+X,4:POKE54273,61:POKE54272,126
265 FORI=15TO0STEP-.1:POKE54296,I:NEXT:POKE54287,0:POKE54286,0:POKESC+Y*40+X,0
270 POKE54273,0:POKE54272,0
275 IFPEEK(7604)=0THENB=B+1:IFB<4THENPOKE54296,15:GOTO210
280 POKEV+24,20:PRINT"JQQQQQQQQQQQQQQQQQQQQQQQQ■                    **FIN DE JUEGO**"
285 PRINT"QPUNTUACION"S:RUN
290 FORAD=7168TO7393
295 READA:POKEAD,A:NEXT
300 FORAD=12288TO12351:READA:POKEAD,A:NEXT
305 POKEV+24,28
310 FORY1=0TO4:FORX1=0TO5:POKE7620+Y1*12+X1*2,Y1*2+1:POKE7621+Y1*12+X1*2,X1*3
315 NEXT:NEXT
320 POKE7605,1:POKE7604,0:SX=-1:SY=0:NO=30:SP=4:RETURN
325 B=1:PRINT"J":FORI=1984TO2023
330 POKEI,5:POKEI+CO,6:NEXT:X=1:Y=23:PE=197:S=0
335 IN=7620:DIMS(5):POKECL+Y*40+X,3:POKESC+Y*40+X,0
340 FORY1=1TO4:PRINT"SQQQQQQQQQQQQQQQQQQQQ"SPC(Y1*10-7)"EEE■■■EEE■■■EEE";
345 PRINT"■■■F G";
350 NEXT:RETURN
355 DATA162,7,142,179,29,162,0,142,180,29,189,197,29,201,255,240,56,141,177,29
360 DATA189,196,29,141,176,29,169,32,141,178,29,32,162,28,189,197,29,24,109
365 DATA181,29,157,197,29,201,0,208,5,160,1,140,180,29,201,39,208,5,160,1,140
370 DATA180,29,141,177,29,169,1,141,178,29,32,162,28,232,232,224,60,208,187
375 DATA173,180,29,208,1,96,169,0,141,180,29,169,32,141,178,29,162,0,189
380 DATA197,29,201,255,240,27,141,177,29,189,196,29,141,176,29,32,162,28,254
385 DATA196,29,189,196,29,201,22,208,5,160,1,140,180,29,232,232,224,60,208
390 DATA216,173,181,29,201,1,240,4,169,1,208,2,169,255
395 DATA141,181,29,173,180,29,208,3,76,0,28,96
400 DATA138,72,152,72,172,176,29,169,0,133,31,169,4,133,32,162,0,24,169
405 DATA40,109,31,0,133,31,169,0,109,32,0,133,32,232,236,176,29
410 DATA208,235,172,177,29,173,178,29,145,31,24,169,212,109,32,0,133
415 DATA32,173,179,29,145,31,104,168,104,170,96,0,16,16,56,124
420 DATA124,254,254,126,90,255,189,189,36,66,36,0,0,0,16,16,16,56,84,0,84,56
425 DATA16,56,56,16,0,108,213,128,132,39,174,197,78
430 DATA255,255,255,255,255,255,255,255,254,252,248,240,224,192,128
435 DATA255,127,63,31,15,7,3,1
```

PROGRAMA COMPROBADO
commodore Magazine

Programas

Othello

REVERSI es el nombre de un juego de mesa de los llamados «de tablero» para dos jugadores, del que mucha gente casi seguro ha oído hablar.

Fue inventado alrededor de 1880 por el londinense **Lewis Waterman.** Más tarde **Jonh W. Mollett** afirmaba que él era el verdadero inventor. Así mientras estos señores se peleaban por adjudicarse la autoría del **Reversi** pasaron los años y en 1971 el japonés **Goro Hasegawa** "reinventó" y rebautizó el juego alterando las cuatro posiciones iniciales y llamándole "OTHELLO".

El juego comienza con cuatro fichas colocadas en el centro de un tablero de 8 × 8 posiciones (dos de cada jugador y cruzadas). Los movimientos se realizan alternativamente colocando cada jugador una ficha junto a otra de su adversario, lo cual sólo es permitido cuando con la pieza forma una o varias líneas en cualquier sentido (vertical, horizontal y diagonal) con otra de su mismo color y encerrando entre ambas a las de su oponente.

Así, todas las piezas que quedan dentro de esas dos, se las da la vuelta incrementándose un marcador y decrementándose otro. Cuando el tablero esté completo, el jugador que tenga más fichas de su color habrá ganado la partida.

En nuestra versión para el **CBM 64,** se organiza una partida entre dos jugadores (no se juega contra la máquina) incluyendo la posibilidad de ver repetidas las jugadas, detenerlas y cambiarlas y también la posibilidad de "pasar".

Las fichas se colocan moviendo el símbolo de **Commodore** (aparecerá parpadeando en la esquina superior izquierda) a la posición requerida usando el *Joystick* en el *Port 2* ó bien, mediante las teclas de movimiento del cursor. Una vez situado, pulsando el botón de fuego o la barra de espacios se sitúa la ficha.

Si intentas hacer un movimiento ilegal, el ordenador lo ignorará.

```
0 REM **************
1 REM **  OTHELLO **
2 REM **************
10 POKE56,56:POKE52,56:CLR
20 GOSUB58000:PRINT"                    EMPEZAMOS? (S O N)          █":GOSUB1420
30 GETR$:IFR$=""THEN30
40 PRINT"▯":IFR$="N"THENEND
50 UD$="████████████████████████"
60 DIMSC(2):DIMTU(1,65):FF=0
99 :
100 REM **** CUADROS TABLERO ****
110 DIMBO$(2)
120 BO$(0)="█QAB███CDE███FGH.T]"
130 BO$(1)="▯IJK███LMN███OPQ.T]"
140 BO$(2)="▯IJK███LMN███OPQ.T]"
199 :
200 REM **** CARACTERES AMPLIADOS ****
210 DIMCH$(7)
220 CH$(0)="')███K*.]":CH$(1)="RT███SU.]"
230 CH$(2)="Z£███[].]":CH$(3)="#%███$&.]"
240 CH$(4)="VX███WY.]":CH$(5)="+-███,..]"
250 CH$(6)="/1███02.]":CH$(7)="35███46.]"
299 :
300 REM **** NUMEROS AMPLIADOS ****
310 NO$(0)=CH$(2):NO$(1)="79███8:.]"
320 NO$(2)=";=███K>.]":NO$(3)="?@███├|.]"
330 NO$(4)="-¬███┌_.]":NO$(5)="|\███||.]"
340 NO$(6)="/\███L/.]":NO$(7)="┌@███┌┘.]"
350 NO$(8)="♦/███|X.]":NO$(9)="o|███♦+.]"
399 :
400 REM ****        SPRITE #0        ****
```

PROGRAMA COMPROBADO commodore Magazine

```
410 POKE2040,13
420 POKEV+28,1:POKEV+37,2:POKEV+38,7:POKEV+39,6
430 POKEV,26:POKEV+1,50
499 :
500 REM ****      SPRITE #1       ****
510 POKE2041,14
520 POKEV+29,2:POKEV+23,2:POKEV+40,14
530 POKEV+2,220:POKEV+3,83
599 :
600 REM **** SELECCION DEL CONJUNTO DE CARACTERES ****
610 POKEV+24,30
699 :
700 REM **** SELECCION MODO MULTICOLOR ****
710 POKEV+22,216:POKEV+34,4:POKEV+35,9
799 :
800 REM **** PREPARA PANTALLA ****
810 GOSUB9010
899 :
999 :
1000 REM ********* BUCLE PRINCIPAL *********
1010 POKES+24,0:F=F+3:IFF>3THENF=0
1020 POKEV+21,F:POKEV,26+24*(XX-1):POKEV+1,50+24*(YY-1)
1030 IFFFTHEN7000:REM REPETICION
1040 IFSC(1)+SC(2)=64THEN6000
1050 GETR$:JO=PEEK(56320)
1060 IF JO=125THENR$="Ж"
1070 IF JO=126THENR$="]"
1080 IF JO=119THENR$="▐"
1090 IF JO=123THENR$="▌"
1100 IF JO=111THENR$=" "
1110 IFR$="Ж"THENYY=YY+1:IFYY>8THENYY=1
1120 IFR$="]"THENYY=YY-1:IFYY<1THENYY=8
1130 IFR$="▐"THENXX=XX+1:IFXX>8THENXX=1
1140 IFR$="▌"THENXX=XX-1:IFXX<1THENXX=8
1150 IFR$=" "ANDBO(XX,YY)<1THENPOKEV+21,2:GOTO1210
1160 IFR$="■"THEN5010
1170 IFR$="Ж"THENFF=1:TU=0
1180 GOTO1010
1199 :
1200 REM **** BUSCA FILAS ****
1210 F1=0
1220 FORDX=-1TO1:FORDY=-1TO1:F=0:IFDX=0ANDDY=0THEN1270
1230 FORI=1TO8:BO=BO(XX+DX*I,YY+DY*I):IFBO=0THENI=8:GOTO1260
1240 IFI=1ANDBO=PLTHENI=8:GOTO1260
1250 IFBO=PLTHENF=1:I=8
1260 NEXTI:IFFTHENGOSUB1310
1270 NEXTDY,DX
1280 IFF1THEN5010
1290 GOTO1010
1299 :
1300 REM **** ACTUALIZA MARCADORES ****
1310 IFF1THEN1330
1320 Y=YY:X=XX:SC(PL)=SC(PL)+1:BO(X,Y)=PL:GOSUB1410
1330 FORJ=1TO8:IFBO(XX+J*DX,YY+J*DY)=PLTHENJ=8:GOTO1380
1340 IFPL=1THENSC(2)=SC(2)-1:GOTO1360
1350 SC(1)=SC(1)-1
```

Programas

Viene de la página anterior

```
1360 SC(PL)=SC(PL)+1:GOSUB1510
1370 X=XX+J*DX:Y=YY+J*DY:GOSUB1410
1380 NEXTJ:F1=1:RETURN
1399 :
1400 REM *** IMPRIME CONTADOR,RUIDO ****
1410 PRINTLEFT$(UD$,(Y-1)*3+1)TAB((X-1)*3)BO$(PL):BO(X,Y)=PL
1420 POKES+5,15:POKES+6,81:POKES+24,15
1430 POKES+4,129:FORT=25TO35:POKES+1,T:POKES,T:NEXT:POKES+4,128
1440 RETURN
1499 :
1500 REM **** IMPRIME TANTEO ****
1510 PRINT"▮▮▮▮▮▮▮▮"TAB(32);:FORI=1TO2
1520 PRINTNO$(INT(SC(I)/10))NO$(SC(I)-INT(SC(I)/10)*10)
1530 PRINT"▮▮▮▮"TAB(32)
1540 NEXT:RETURN
1599 :
5000 REM **** CAMBIÓ DE JUGADOR ****
5010 IFPL=1THENPL=2:POKEV+3,125:GOTO5030
5020 PL=1:POKEV+3,83
5030 TU(0,TU)=XX:TU(1,TU)=YY:TU=TU+1:XX=1:YY=1:GOTO1010
5999 :
6000 REM **** FIN DEL JUEGO ****
6010 POKEV+21,0
6020 FORI=1TO50
6030 POKEV+35,6:FORJ=0TO50:NEXT
6040 POKEV+35,14:FORJ=0TO50:NEXT
6050 NEXT
6060 POKEV+24,21:POKEV+33,0:POKEV+32,0:PRINT"▮▮▮▮▮"
6070 PRINTSPC(14)"************"
6080 PRINTSPC(14)"*    FIN    *"
6090 PRINTSPC(14)"************"
6100 PRINT"▮▮▮▮     JUGADOR UNO (AZUL):  ";SC(1);" PUNTOS"
6110 PRINT"▮▮▮▮     JUGADOR DOS (BLANCO):  ";SC(2);" PUNTOS"
6120 PRINTSPC(14)"▮▮▮▮▮▮OTRA PARTIDA?"
6130 CLR:S=54272:V=53248:GOTO30
6999 :
7000 REM ****   REPETICION   ****
7010 IFTU>0THEN7050
7020 GOSUB9000
7030 PRINT"▮"LEFT$(UD$,18)TAB(29)CH$(5)CH$(0)CH$(4)
7040 PRINT"▮▮"TAB(27)CH$(7)" "CH$(2)" "CH$(6)
7050 XX=TU(0,TU):YY=TU(1,TU)
7060 POKEV,26+24*(XX-1):POKEV+1,50+24*(YY-1)
7070 POKEV+21,3:GETR$:IFR$<>"£"ANDR$<>"←"THENPOKEV+21,0:GOTO7070
7080 POKEV+21,0
7090 IFR$="£"THEN7110
7100 IFTU(0,TU+1)<>0THEN1200
7110 FF=0
7120 PRINTLEFT$(UD$,17):FORI=0TO4:PRINTTAB(27)"     ":NEXT
7130 GOTO1010
7999 :
8000 REM **** COPIA CARACTERES ****
8010 FORBY=0TO7:X=0
8020 FORBI=0TO7
```

```
8030 IF(PEEK(53248+CH*8+BY)AND2↑BI)THENX=X+2↑(BI*2)+2↑(BI*2+1)
8040 NEXTBI
8050 POKEA+BY*2,X/256
8060 POKEA+BY*2+1,X/256
8070 POKEA+16+BY*2,X-(INT(X/256)*256)
8080 POKEA+16+BY*2+1,X-(INT(X/256)*256)
8090 NEXTBY:A=A+32:IFA=14576THENA=A+40
8100 RETURN
8999 :
9000 REM **** RECOLOCA TABLERO ****
9010 FORI=1TO8:FORJ=1TO8:BO(I,J)=0:NEXTJ,I
9020 BO(4,4)=2:BO(5,5)=2:BO(4,5)=1:BO(5,4)=1
9030 POKEV+33,14:POKEV+32,14:PRINT"♥";
9040 FORI=1TO8:FORJ=1TO8:PRINTBO$(BO(I,J));:NEXT:PRINT"▓▓":NEXT
9050 PRINT"▓▓▓▓"TAB(27):PRINT
9060 PRINT"▓▓▓▓"TAB(27)BO$(1)
9070 PRINT"▓▓▓"TAB(27)BO$(2)
9080 SC(1)=2:SC(2)=2:XX=1:YY=1:PL=1:TU=1
9082 GOSUB1510
9090 RETURN
9999 :
58000 REM ****     TITULO      ****
58010 V=53248:S=54272:POKEV+33,11:POKEV+32,11:PRINT"♥▶";
58020 FORI=0TO5:FORJ=0TO19:PRINT" |";:NEXTJ:FORJ=0TO19:PRINT"-+";:NEXTJ,I
58030 FORI=0TO19:PRINT" |";:NEXT
58040 PRINT"▓▓▓▓▓▓▓▓▓ ▓▓▓▓  ▓▓▓▓▓▓▓▓▓▓▓ ▓▓▓▓▓▓▓▓▓▓▓▓▓ ▓▓▓▓";
58050 PRINT"▓▓ ▓▓▓ ▓▓▓▓▓▓ ▓▓▓ ▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓ ▓";
58060 FORI=0TO1:PRINT"▓▓ ▓▓▓ ▓▓▓ ▓▓▓ ▓▓▓ ▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓ ▓";:NEXT
58070 PRINT"▓▓ ▓▓▓ ▓▓▓ ▓▓▓ ▓ ▓▓▓ ▓ ▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓ ▓";
58080 FORI=0TO2:PRINT"▓▓ ▓▓▓ ▓▓▓ ▓▓▓ ▓▓▓ ▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓▓ ▓▓▓ ▓";:NEXT
58090 PRINT"▓▓▓▓ ▓▓▓▓▓▓▓▓▓▓▓▓ ▓▓▓▓ ▓▓▓▓ ▓▓ ▓▓ ▓▓";
58100 PRINT"▓▓DOS JUGADORES AZUL Y BLANCO,PONEN FICHAS";
58110 PRINT"EN EL TABLERO.(USANDO ▓ESPACIO▓ O ▓FUEGO▓ )"
58120 PRINT"     SI LA PIEZA FORMA UNA LINEA (HORIZ.) ";
58130 PRINT"VERT. O DIAG.) CON OTRA,ENTONCES TODAS"
58140 PRINT"LAS FICHAS 'ATRAPADAS' SE DAN LA VUELTA"
58150 PRINT"Y EL MARCADOR CAMBIA DE ACUERDO A ELLO.▓"
58160 PRINT"▓▓ USA LAS TECLAS DEL CURSOR Y/O JOYSTICK"
58170 PRINT"▓      PULSA ▓F8▓ PARA NO HACER LA JUGADA"
58180 PRINT"▓EN EL MODO REPETICION ▓(CTRL' C')▓ USA:"
58190 PRINT"    '▓←▓' PARA SEGUIR , '▓£▓' PARA VOLVER"
58200 PRINT"▓LOS GRAFICOS TARDAN 100 SEG. EN HACERSE▓"
58210 IFPEEK(896)=255THENRETURN
58999 :
59000 REM **** PREPARA GRAFICOS ****
59010 CH=1000:A=14336
59020 READD:IFD>255THENGOSUB59500:GOTO59020
59030 IFD<0THEN59100
59040 CH=CH+D:POKEA,D:A=A+1:GOTO59020
59100 POKE56334,PEEK(56334)AND254
59110 POKE1,PEEK(1)AND251
59120 FORI=0TO16:READCH:GOSUB8000:NEXTI
59130 POKE1,PEEK(1)OR4
59140 POKE56334,PEEK(56334)OR1
```

Programas

Viene de la página anterior

```
59199 :
59200 REM **** BORRA ESPACIO ****
59210 FORI=0TO7:POKE14336+32*8+I,0:NEXT
59299 :
59300 REM ****    SPRITE#0    ****
59310 CH=1000:A=832
59320 READD:IFD>255THENGOSUB59500:GOTO59320
59330 IFD<0THEN59400
59340 CH=CH+D:POKEA,D:A=A+1:GOTO59320
59399 :
59400 REM ****    SPRITE#1    ****
59410 FORI=0TO62:POKE896+1,255:NEXT
59420 RETURN
59499 :
59500 REM **** RUTINA DE COMPROBACION DE DATAS ****
59510 IFD=CHTHENCH=1000:RETURN
59520 PRINT"ERROR DE DATAS EN LA LINEA "PEEK(64)*256+PEEK(63)
59530 PRINT"REPASA EL LISTADO■":END
59600 :
59700 :
60000 REM**** DATAS PARA GRAFICOS TABLERO ****
60001 DATA170,170,149,149,149,149,149,149,2234
60002 DATA170,170,85,85,85,85,85,85,1850
60003 DATA170,170,86,86,86,86,86,86,1856
60004 DATA149,149,149,149,149,149,149,149,2192
60005 DATA85,85,85,85,85,85,85,85,1680
60006 DATA86,86,86,86,86,86,86,86,1688
60007 DATA149,149,149,149,149,149,170,170,2234
60008 DATA85,85,85,85,85,85,170,170,1850
60009 DATA86,86,86,86,86,86,170,170,1856
60010 DATA170,170,149,151,159,159,159,191,2308
60011 DATA170,170,85,255,255,255,255,255,2700
60012 DATA170,170,86,214,246,246,246,254,2632
60013 DATA191,191,191,191,191,191,191,191,2528
60014 DATA255,255,255,255,255,255,255,255,3040
60015 DATA254,254,254,254,254,254,254,254,3032
60016 DATA191,159,159,159,151,149,170,170,2308
60017 DATA255,255,255,255,255,85,170,170,2700
60018 DATA254,246,246,246,214,86,170,170,2632,-1
60099 :
60100 REM**** CARACTERES AMPLIADOS ****
60101 DATA3,5,15,18,19,21,28,31,49,50,51,52,53,54,55,56,57
60199 :
60200 REM**** DATA DE LOS SPRITES ****
60201 DATA0,0,0,0,12,0,3,252,1267
60202 DATA0,15,240,0,63,0,0,60,1378
60203 DATA0,0,240,42,160,240,42,128,1852
60204 DATA240,42,0,240,0,0,240,0,1762
60205 DATA0,240,0,0,240,21,0,240,1741
60206 DATA21,64,240,21,80,252,21,80,1779
60207 DATA60,0,0,63,0,0,15,240,1378
60208 DATA0,3,252,0,0,12,0,1267,-1
```



Programas
/Montajes/Còdigo Màquina
YA ESTA A LA VENTA
AÑO 1 - No. 12 / NOVIEMBRE - 84 - 250 Ptas.
ZX
REVISTA PARA LOS USUARIOS
DE ORDENADORES SINCLAIR
SE-TSR
Novedad
PRESENTACION DEL
SPECTRUM PLUS
CURSO DE FORTH

GUIA PRACTICA



DEFOREST
MICROINFORMATICA

TODO SOBRE **COMMODORE - 64** Y **VIC - 20**

LOS ULTIMOS JUEGOS EN EL MERCADO
TODO EN PERIFERICOS - LIBROS
PROGRAMAS DE GESTION - ETC.

SOLICITE INFORMACION POR CORREO

BARCELONA-15
C/ Viladomat, 105. Tel. 223 72 29



PRECIOS ESPECIALES PARA COLEGIOS Y TIENDAS

VIC-20
COMMODORE 64
ZX81 1K
SPECTRUM 48K
ORIC ATMOS 48K
MICRODRIVE
INTERFACE
JUEGOS (Importados)

* * *

Tel. (93) 242 80 11 BARCELONA
Tel. (93) 319 39 65 BARCELONA
Tel. (93) 725 20 59 SABADELL
(A partir 18.00 horas)

* * *

MICRO / RAM
Obispo Laguarda 1, 1.º
08001 BARCELONA

COMMODORE 64
SPECTRUM
CASIO

PRECIOS INTERESANTES

CMP

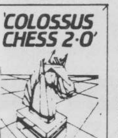
Arturo Soria, 154
Tel. 415 93 28
28043 MADRID

CENTRO DE INFORMATICA
Las Rozas - Majadahonda

EMPEZAMOS
Cursillos en BASIC
cada 15 días

Directamente con ordenadores
VIC-20 COMMODORE 64
SPECTRAVIDEO

Tfno. 637 31 51



COLOSSUS CHESS 2-0
El mejor juego de ajedrez creado hasta el momento en su nivel superior, supera a todos los demás programas de ajedrez en otros ordenadores; tardaron siete años en crearlo, pero al fin lo tenemos aquí con manual en castellano.
Por sólo 2.750 ptas.

Precios especiales para distribuidores

Todos los simuladores de vuelo para el COMMODORE

COMMODORE-64
Instrucciones en castellano
Flight Part 737 — 1.950 pts.
Pilot 64 — 1.950 pts.
Solo Flight — 3.850 pts.
Flight Sim.II — 13.900 pts.
(resumen en cast.)

VIC 20
Flight Part 737 — 1.900 pts.
Instrucciones en castellano

ASTOC-DATA
Hardware y Software-Systems
Sarela de Abajo
Santiago de Compostela
Tel. 599533

Precios especiales para distribuidores

ANUNCIESE por MODULOS

MADRID
(91) 733 96 62
BARCELONA
(93) 301 47 00

ULTIMO AVISO

¿Eres aficionado a la programación?
¿Dominas el código máquina?
¿Tienes programas originales?
¿Puedes escribir un buen juego?
¿Quieres ganar dólares, libras, francos o pesetas desde tu casa, en tus horas libres?

¡NO TE LO PIERDAS!

Contacta inmeditamente con:

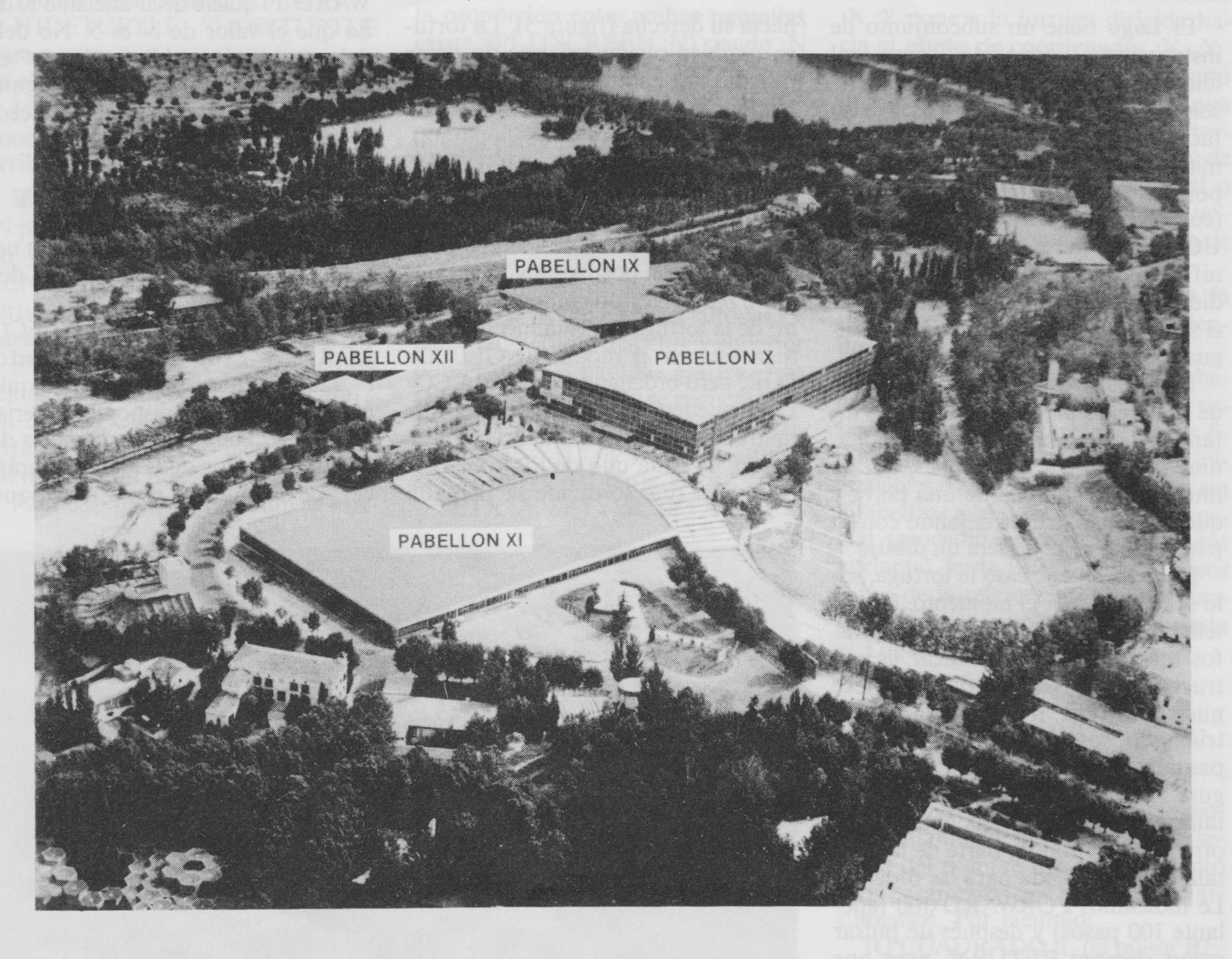
CIBERCOMP, S. A.
Tels. (91) 200 21 00
(91) 759 22 44

Especialistas en software para Home Computers, asociados con primeras firmas internacionales.




Del 16 al 23 de Noviembre de 1984

ESTAREMOS EN EL SIMO'84



Stand-D-168
PABELLON XI

Stand-E-14
PABELLON XII

SEGUNDA PARTE
Logo para el C-64

El **Logo** tiene un subconjunto de instrucciones, conocido como «el idioma de la tortuga», que permite resolver problemas geométricos. Por su facilidad de uso en gráficos, las primitivas fundamentales han sido incorporadas en ampliaciones de BASIC (como **ULTRABASIC**) o en el **Pascal UCSD.** Aunque es una parte muy significativa y, sin duda, la más difundida del **Logo,** no se debe identificar el sublenguaje de la tortuga con el lenguaje **Logo.**

La tortuga ofrece un gran campo de aplicación para programadores infantiles. «Imaginemos una tortuga, muy obediente a las órdenes, que se mueve sobre la arena de una playa y que, al trasladarse, va dejando con su panza un rastro que será un dibujo.»

El **Logo,** en este caso la tortuga, sólo sabe inglés por el momento. El castellano se lo enseñaremos dos párrafos más adelante. Al indicar la instrucción DRAW (dibujar) aparece nuestra tortuga, representada por un triángulo isósceles, en el centro de la pantalla y orientada hacia arriba (figura 3). Se ve a la tortuga en la pantalla gráfica y también aparece, en otro color de fondo, parte de la pantalla de texto usada para las órdenes. Le indicamos FORWARD 100 (adelante 100 pasos) y después de pulsar como siempre RETURN, para que «nos oiga», vemos que se mueve y queda dibujado un segmento de longitud 100 (figura 4). A continuación tecleamos RIGHT 90 (derecha 90 grados) y se observa que la tortuga no avanza sino que gira un ángulo recto hacia su derecha (figura 5). La tortuga también puede moverse sin dejar rastro: para ello debe ordenarse PENUP(=levanta la pluma). Lo contrario es PENDOWN (=baja la pluma). Si pulsamos PENUP RIGHT 45 FORWARD 100 veremos la figura 6.

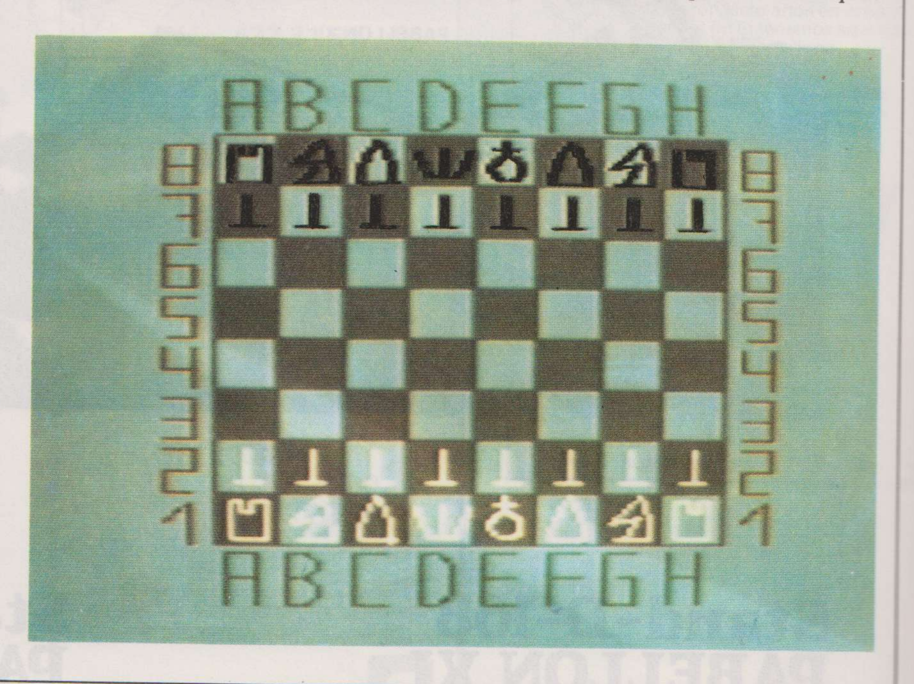
La tortuga se mueve en un entorno de alta resolución, con 320*200 puntos, donde son posibles los 16 colores tanto para el fondo como para el rastro de la tortuga. Normalmente la tortuga está en el modo SINGLECOLOR, pero ordenando DOUBLECOLOR pueden coexistir 2 colores distintos para puntos, aparte del color de fondo, en cada uno de los 40*25 espacios en que se divide la pantalla gráfica, a cambio la resolución baja a 160*200.

Dado que en **Logo** cualquier letra o palabra significa un procedimiento, las variables (numerales, literales o listas) deben diferenciarse para evitar la confusión. Cuando se define su valor se anteponen unas comillas al nombre de la variable. Asi: MAKE "N 5 hace que la variable "N valga 5. Cuando nos referimos al valor (no al nombre) de la variable, debe aparecer precedida por dos puntos. Así: FORWARD :N quiere decir adelante 5, dado que el valor de N es 5. No debe dejarse espacio en blanco entre " o : y la variable. Si N aparece así significa procedimiento N (como procedimientos eran AJEDREZ, ...).

INSTRUCCIONES GRAFICAS

Las primitivas gráficas forman una colección numerosa, y podemos destacar las siguientes:

SINGLECOLOR, DOUBLECOLOR. Si los 64.000 puntos (=320*200) pudiesen ser de cualquiera de los 16 colores disponibles, serían precisos 32.000 Bytes de memoria dedicados al color de la pantalla gráfica. Agrupando los *pixels* en bloques



de 8*8 (el espacio de un carácter en la pantalla de texto) y permitiendo a los sumo 2 colores en cada bloque, aparte el del fondo, solamente se requiere 1 KB.

FORWARD :N, BACK :N = adelante, atrás N pasos en el sentido hacia el que apunta la tortuga.

RIGHT :N, LEFT :N = gira a la derecha, izquierda N grados sexagesimales.

PENDOWN, PENUP baja o suble la pluma para dejar o no rastro.

HIDETURTLE SHOWTURTLE para ocultar o mostrar la tortuga, que es el sprite 0.

HOME lleva a la tortuga al centro, coordenadas 0,0, y la orientada hacia arriba.

BACKGROUND :N, PENCOLOR :N para dar un color del 0 al 15 al fondo y a la tortuga (y a su rastro si PENDOWN). Si PENCOLOR —1 la tortuga borra la trayectoria por la que pasa (es equivalente a PENERASE).

STAMPCHAR :N permite representar caracteres en la pantalla gráfica. STAMPCHAR "A dibujará una letra A. Solamente es bien legible en SINGLECOLOR.

Para casi todas las primitivas existen formas abreviadas, tales como FD :N = FORWARD :N, ...

La pantalla de texto puede tener definidos los colores del fondo y de los caracteres mediante TEXTBG :N y TEXTCOLOR :N. Para presentar únicamente la pantalla gráfica o de texto se emplean FULLSCREEN o TEXTSCREEN. Para combinarlas SPLITSCREEN, que presenta las 10 líneas inferiores de la pantalla de texto, con sus colores característicos, según se hayan establecido. Para variar la proporción entre ambas pantallas existe SPLITSCREEN :N) siendo :N el número de líneas de texto a mostrar.

CLEARSCREEN y CLEARTEXT borran respectivamente las pantallas gráfica y de texto.

DRAW (dibujar) y NODRAW (no dibujar) pasan al modo gráfico o de texto.

WRAP (envolver, enrrollar) y NOWRAP permiten conectar o no los lados izquierdo y derecho, y superior o inferior de la pantalla gráfica. Tras DRAW si si la tortuga se encuentra en estado WRAP y desaparece por arriba, reaparecerá por abajo (si estuviese en NOWRAP daría error).

SETX :A, SETY :B, SETXY :N :M pone a la tortuga en la abscisa :A, o en la ordenada :B o en las coordenadas :N, :M.

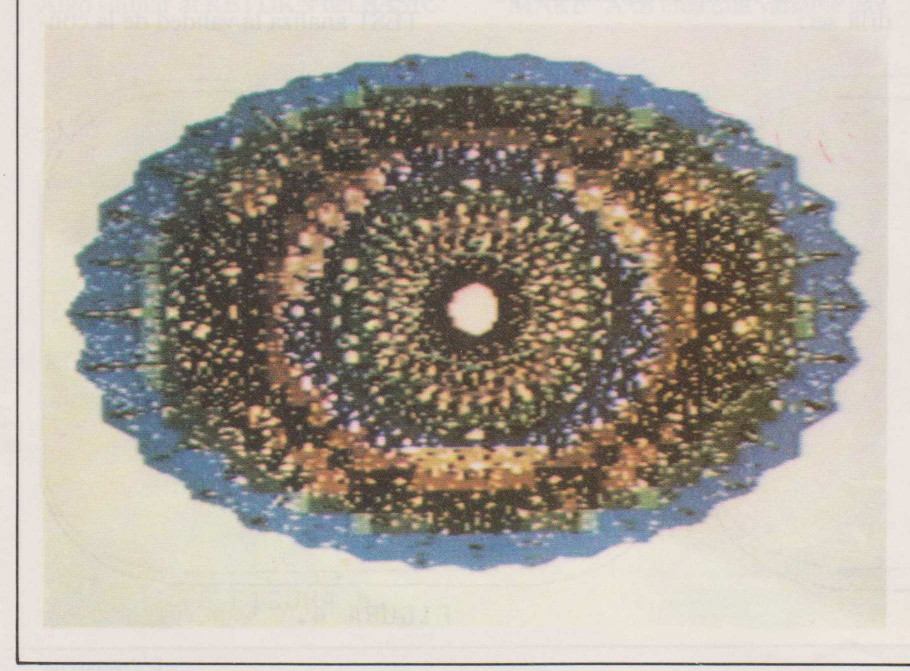
XCOR (coordenada x) e YCOR indican la abscisa y la ordenada en la que está la tortuga.

HEADING (orientación) da el ángulo en el que mira la tortuga, de 0 a 360 grados. 0 es hacia arriba, 90 hacia la derecha, ...SETHEADING :N pone a la tortuga en la dirección :N.TOWARDS :X :Y utilizado de la forma SETHEADING TOWARDS :X :Y pone a la tortuga dirigida hacia el punto de coordenadas :X, :Y.

DRAWSTATE da como resultado una lista conteniendo las condiciones de dibujo: pluma arriba o abajo; tortuga visible o no; color de fondo y de rastro; modo DRAW o no; multi o monocolor; pantalla gráfica, de texto o mezcladas; color del fondo y de los caracteres del texto.

Otras sentencias primitivas de la tortuga que se refieren a los «duendes» se explican más adelante.

Las posibilidades del **Logo** para hacer todo tipo de dibujos, tanto artísticos como geométricos (véase las figuras 7, 8 y 9 como muestra) son inigualables. Desde niños de 4-5 años sin alfabetizar aún o en Educación Especial (procedimiento INSTANT más adelante), hasta físicos y matemáticos (libro de la referencia 4) han utilizado la herramienta gráfica que brinda el **Logo.**



DEFINICION Y EDICION DE PROCEDIMIENTOS

Para crear una instrucción se emplea usualmente la primitiva TO, seguida del nombre del procedimiento y los parámetros que utilice como argumentos. Por ejemplo:

TO CUADRADO :L (al pulsar RETURN se pasa al modo Edición).

FD :L RT 90 FD :L RT 90 FD :L RT 90 FD :L RT 90 (se recuerda que FD = FORWARD y RT = RIGHT).

Se teclea CONTROL-C o la tecla STOP y al salir del modo Edición aparece en la pantalla de texto

«PLEASE WAIT...» (Por favor, espere) y casi instantáneamente CUADRADO DEFINED. Ahora basta pulsar CUADRADO 40 para que la tortuga avance 40 pasos, gire 90 grados a la derecha, avance... y dibuje un cuadrado de 40 pasos de lado. La nueva orden CUADRADO queda almacenada en el «espacio de trabajo» y no se borrará hasta que ordenemos GOODBYE, sería aconsejable guardar en *diskette* los procedimientos creados, si son interesantes. Con SAVE "NOMBRE.DEL.FICHERO se deposita en el disco de trabajo todos los procedimientos definidos y las variables globales empleadas con sus valores. Para recuperar los programas, que no son sino superprocedimientos, se emplea READ "NOMBRE.DE.FICHERO. Si algún fichero grabado ya no es útil ERASEFILE "NOMBRE.DE.FICHERO lo borrará.

Si se quiere anular un procedimiento innecesario del espacio de trabajo, para disponer de más memoria de nodos, basta ordenar ERASE "NOMBRE (por ejemplo ERASE "CUADRADO). Para volver al modo Edición se emplea EDIT, lo que permite modificar el último procedimiento. Asimismo en todo momento podemos repetir TO CUADRADO y nos aparecerá en el modo Edición la definición anterior para reformar si fuera preciso.

Otra forma de definir procedimientos, menos cómoda, es mediante la instrucción DEFINE. El ejemplo anterior se hubiese podido realizar así: DEFINE "CUADRADO [[:L][FD :L RT 90 FD :L RT 90 FD :L RT 90 FD :L RT 90]]. DEFINE toma el nombre del procedimiento (precedido de ") y una lista (entre corchetes), siendo el primer elemento a su vez una lista con las variables del procedimiento (si no las tuviere, la lista debe aparecer como []).

La primitiva TEXT "NOMBRE nos muestra en forma de lista el procedimiento "NOMBRE, que ha debido ser definido anteriormente.

¡HABLEMOS CASTELLANO!

Traducir externamente las instrucciones primitivas en inglés a procedimientos en castellano es trivial, según se deduce de lo expuesto. Por ejemplo, mediante: TO ADELANTE :N
FD :N
y pulsando la tecla STOP el ordenador ya entiende la orden ADELANTE :N. Su forma compendiada podría ser:

TO AD :N
ADELANTE :N
y así crear un fichero que titulado CASTELLANO y pasado a *diskette* con SAVE "CASTELLANO nos serviría para que otro día tras cargar el **Logo** y leer este archivo con READ "CASTELLANO, dispongamos ya de **Logo** en español.

Se rumorea que **Commodore** está preparando una versión con primitivas en castellano, lo que sería preferible por normalizar la nomenclatura y para reservar el espacio de trabajo para otros menesteres.

PRIMITIVAS CONDICIONALES, DE CONTROL Y DE LAS VARIABLES

Se enumeran a continuación otras sentencias primitivas, con una sumaria explicación. Se recuerda que las principales órdenes tienen abreviaturas que no se indican en este resumen.

Condicionales:

NOT, IF...THEN ...ELSE como en BASIC (el THEN puede suprimirse).

ALLOF, ANYOF equivalentes a AND y OR.

TEST analiza la validad de la con-

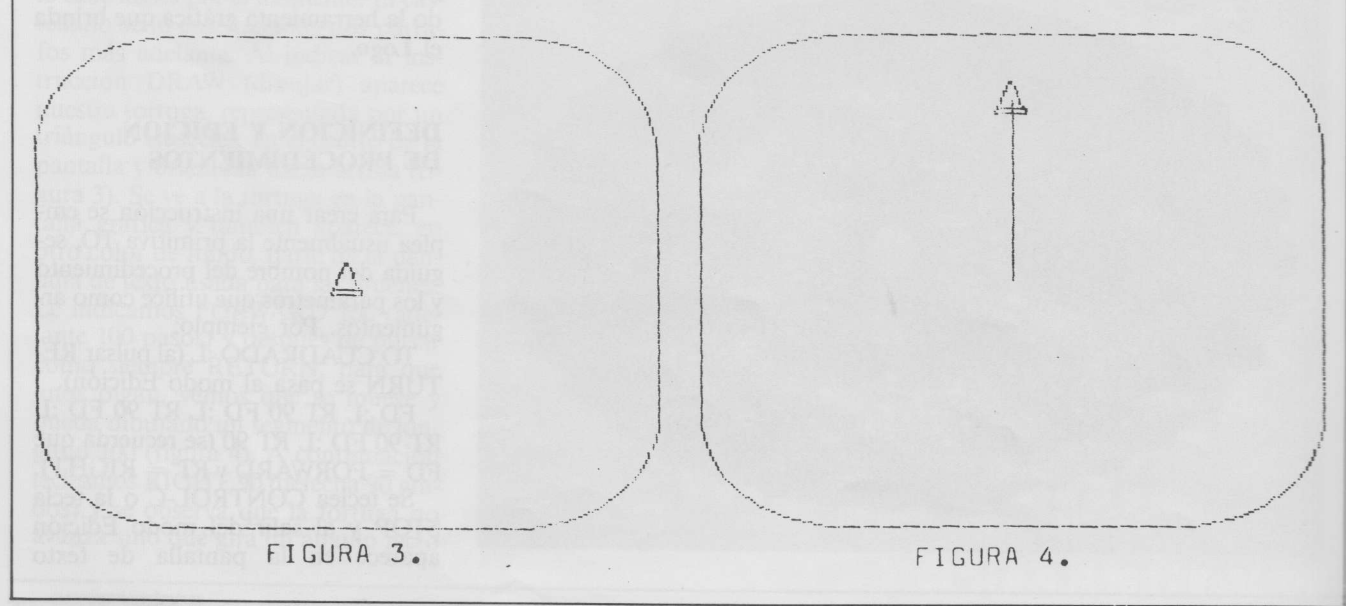


FIGURA 3.   FIGURA 4.

dición y da TRUE (verdadero) o FALSE.

IFTRUE, IFFALSE ejecuta el resto de la línea según el resultado del TEST precedente.

De control:

GO (procedimiento) realiza el procedimiento. Es superflua (como LET en BASIC y basta teclear el nombre del procedimiento. Se utiliza ocasionalmente para mayor claridad.

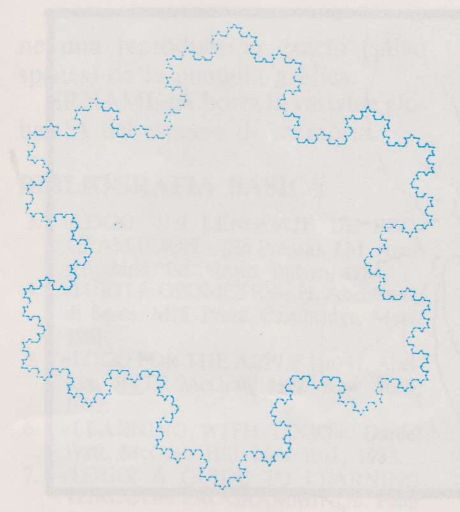
GOODBYE o pulsando simultáneamente las teclas STOP y RESTORE, para limpiar el espacio de trabajo y restablecer el **Logo** inicial, pero no borra el área de lenguaje máquina del usuario.

OUTPUT efectúa la evaluación de un procedimiento y al obtener un resultado se detiene en ese punto y envía el valor al procedimiento precedente.

RUN :N toma como argumento una lista :N y la ejecuta paso a paso como si fuese tecleada.

REPEAT :N :L ejecuta como RUN la lista :L tantas veces como indica el valor :N. El procedimiento CUADRADO :L podría ser simplemente REPEAT 4 [FD :L RT 90].

STOP detiene la ejecución del procedimiento en el que aparece y devuelve el control al procedimiento previo. Algo similar al RETURN del BASIC.



TOPLEVEL (o pulsando CONTROL-G) interrumpe la ejecución de todo el programa. (END se utiliza en la definición de procedimientos y es colocado automáticamente cuando se sale del modo Edición).

Relativas a las variables:

LOCAL "A hace local a la variable A en la ejecución de un procedimiento y no mantiene su valor después (con el consiguiente ahorro de memoria). Las variables que son argumentos de un procedimiento tienen este carácter siempre.

MAKE "A :B crea una variable global (a menos que sea antecedida por LOCAL "A) y le asigna el valor :B (número, palabra o lista, que a su vez puede ser de números, palabras o listas...).

THING "A equivale a :A.

THING? "A da como resultado TRUE (verdadero) si la variable "A tiene un valor asignado.

CALCULO

Aparte de los operadores y las funciones comunes con el BASIC como =, +, —, *, /, > , < , SIN y COS (operan con grados, no con radianes), SQRT (raíz cuadrada, como SQR en BASIC), RANDOM, RANDOMIZE, INTEGER (como INT), QUOTIENT y REMAINDER (cociente y resto enteros, como DIV y MOD), se dispone de las siguientes instrucciones:

ROUND :A redondea al entero más cercano (ROUND 4.8 = 5, INTEGER 4.8 = 4).

NUMBER? :A da TRUE si la variable "A tiene asociado un número como valor, no una palabra ni una lista.

ATAN :A :B da en grados el arco de :A/:B.

BITAND, BITOR, BITXOR toman como argumentos 2 enteros y dan co-

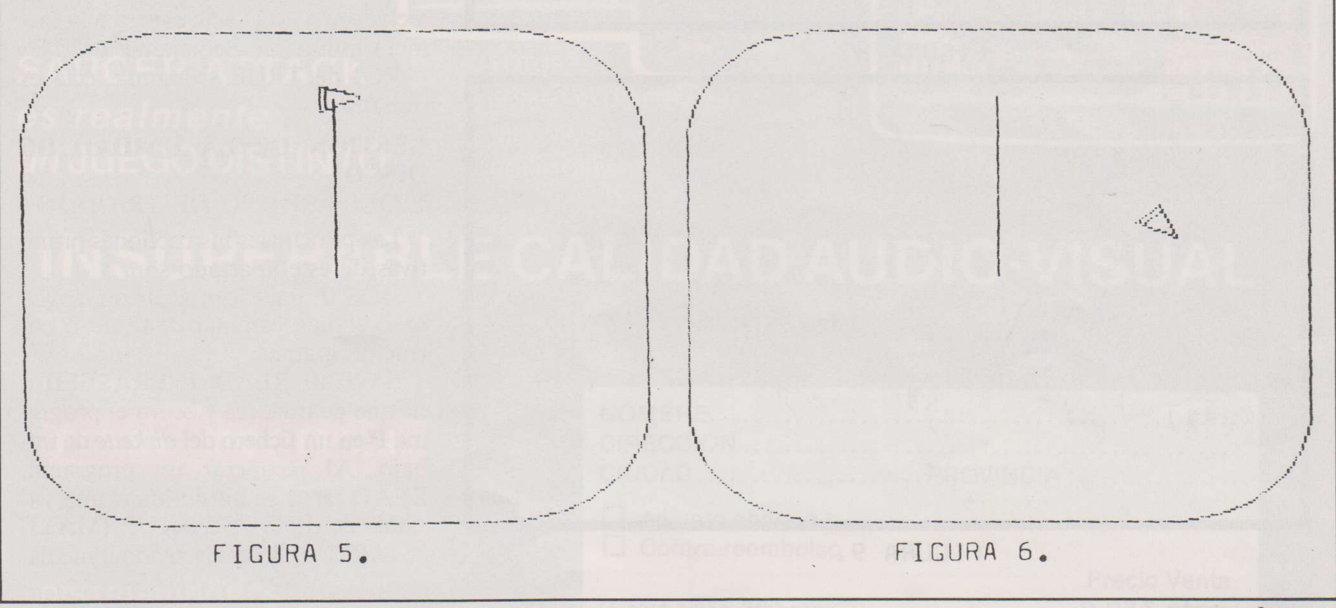


FIGURA 5.   FIGURA 6.

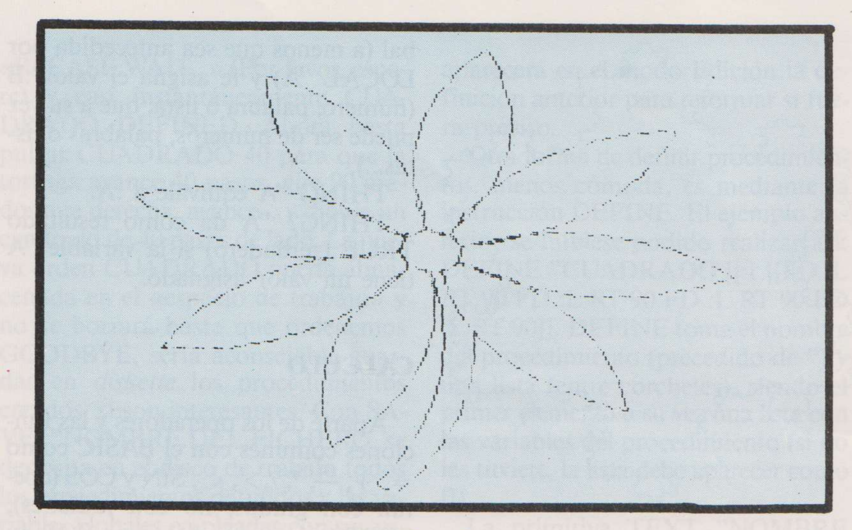


FIGURA 7.

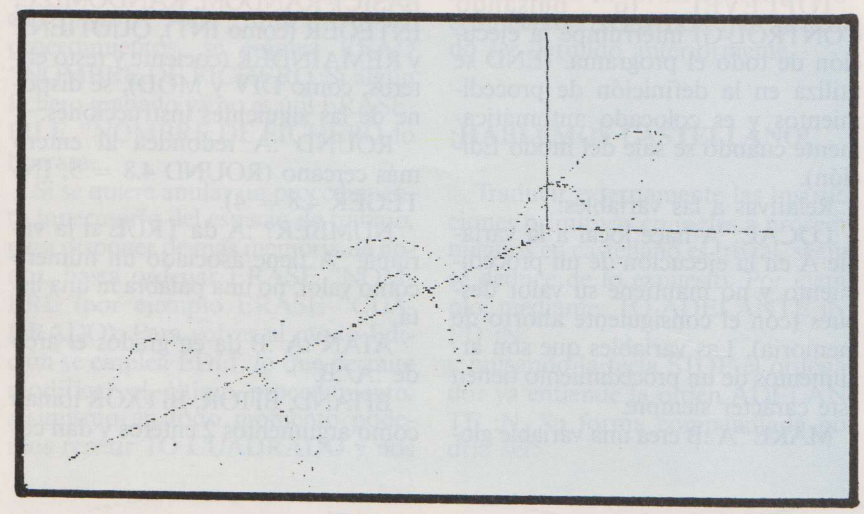


FIGURA 8.

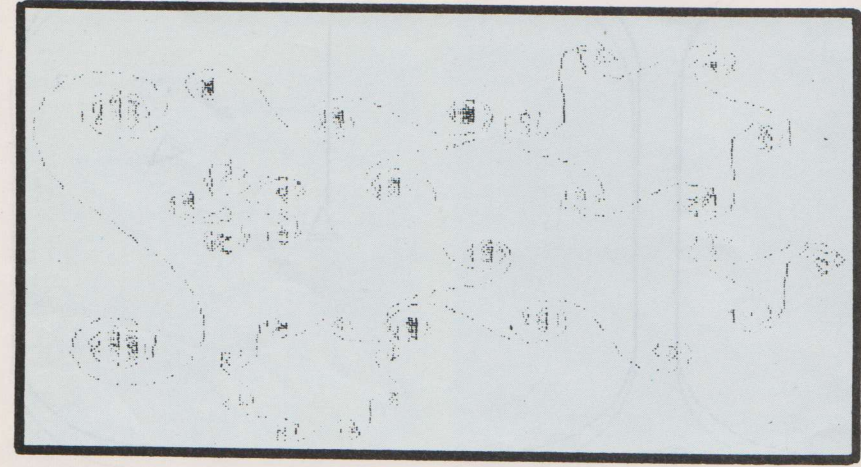


FIGURA 9.

mo resultado su composición booleana mediante el conector lógico AND, OR y OR exclusivo, respectivamente. Ejecutan bit a bit las operaciones lógicas AND, OR y XOR como en BASIC.

ENTRADA-SALIDA

Las principales primitivas son:

ASCII :N Y CHAR :M, la primera toma un carácter como dato y da su número de códigos ASCII y la segunda es su recíproca. Como ASC y CHR$ en BASIC.

CURSOR :C :F coloca el cursor en la columna :C y fila :F. Se complementa con CURSORPOS, que indica la posición actual del cursor. Como PRINT AT en BASIC.

JOYSTICK :N y JOYBUTTON :N, PADDLE :N y PADDLEBUTTON :N para control de estos periféricos de entrada, muy útiles en juegos de animación, especialidad del **Logo.**

PRINTER y NOPRINTER para activar y desactivar la impresora.

PRINT :N,PRINT1 :N,FPRINT :N para presentar resultados en distintas formas.

CLEARINPUT limpia el *buffer* del teclado.

REQUEST recoge una entrada en forma de lista tras pulsar RETURN. Similar al INPUT del BASIC.

READCHARACTER da como resultado el primer carácter de *buffer* de teclado y si está vacío espera a la primera pulsación. Semejante a GET.

RC? da TRUE si alguna tecla es pulsada.

GESTION DE LA UNIDAD DE DISCO Y DEL ESPACIO DE TRABAJO

Las principales instrucciones primitivas de este apartado son:

DOS :L para formatear un *diskette*, o copiar, cambiar de título o borrar programas.

SAVE :P, READ :P, ERASEFILE :P que guarda, lee y borra el programa P en un fichero del *diskette* de trabajo. Al recuperar un programa, READ :P, si se ha definido una variable de título "STARTUP (MAKE "STARTUP :X) es ejecutado inmedia-

tamente su contenido, que debe ser una lista, como un RUN :X.

CATALOG muestra los programas disponibles en el disco. Los títulos aparecen seguidos de los sufijos .LOGO, .PIC1, .PIC2 o .SHAPES según contengan un programa, el dibujo y color de la copia de una pantalla gráfica o la forma de un «duende» (ver más adelante). Los archivos en código máquina cambian el sufijo.

PRINTOUT :A indica la definición del procedimiento :A. Puede utilizarse con ALL, PROCEDURES, NAMES o TITLES, resultando la definición de todos los procedimientos y valores de las variables, o sólo los procedimientos, o sólo las variables, o únicamente los títulos de procedimientos, almacenados en el espacio de trabajo.

POTS = PRINTOUT TITLES.

SAVEPICT :D, READPICT :D, ERASEPICT :D que salva, lee o borra un archivo, titulado D que contiene una reproducción exacta (salvo sprites) de la pantalla gráfica.

ERNAME :A borra la variable global :A del espacio de memoria.

BIBLIOGRAFIA BASICA

3. «LOGO: UN LENGUAJE DE PROGRAMACION». (En Prensa). J.M. Aguirregabiria. Ed. Urmo. Bilbao, 1984.
4. «TURTLE GEOMETRY». H. Abelson-A. di Sessa. MIT Press. Cambridge, Mass. 1981.
5. «LOGO FOR THE APPLE II». H. Abelson. BYTE/McGraw Hill. New York, 1982.
6. «LEARNING WITH LOGO». Daniel Watt. McGraw Hill. New York, 1983.
7. «LOGO: A GUIDE TO LEARNING THROUGH PROGRAMMING». Peter Goodyear. Ellis Horwood. London, 1983.
8. «DISCOVERING APPLE LOGO». Daniel Thornburg. Addison-Wesley. California, 1983.
9. «AN INTRODUCTION TO LOGO». J. Dale Burnett. Creative Computing Press. N. Jersey, 1982.
10. «APPLE, LOGO PRIMER». Gary Bitter. Prentice Hall, 1983.
11. «UNDERSTANDING LOGO». Richard Peddicord. Alfred Publishing Co., 1983.
12. «LOGO PROGRAMING». Peter Ross. Addison Wesley, 193.
13. «LEARNING LOGO IN THE APPLE II». Anne McDougall. Prentice Hall of Australia, 1982.
14. «1, 2, 3 MY COMPUTER & ME». D. Bearden. Prentice Hall, 1983.
15. «THE TURTLE'S SOURCEBOOK». D. Bearden. Prentice Hall, 1983.
16. «PREMIERES PAS EN LOGO». Marie-Gaelle Monteil. Eyrolles. París, 1984.
17. «LOGO». Gerard Weidenfeld. Eyrolles. París, 1984.

Los artículos sobre Logo en revistas son muy abundantes. Se pueden citar:

18. «VOYAGE AU CENTRE DE LOGO». Pierre Camilhe. L'Ordinateur Individuel. Octubre de 1983. Traducido por El Ordenador Personal.

JOSE MIGUEL AGUIRREGABIRIA,

Profesor titular de la Escuela Universitaria de Profesorado de EGB de Bilbao. Coordinador del equipo de EAO de Vizcaya.
Avda. Santa Ana, 10-1. C. Leioa (Vizcaya).
Teléfono: (94) 464 19 97.

Construye

La principal ventaja que aporta él *paddle* es la posibilidad de conseguir una gama de valores escalonados, que podemos aplicar a nuestro programa para ejercer un control más exacto sobre un determinado proceso de lo que hace posible el *joystick*. Con él, estando en posesión de un programa de minucioso desarrollo, el funcionamiento puede ser más intuitivo para el usuario que si empleamos un *joystick*.

Cada uno de los dos accesorios tiene sus ventajas e inconvenientes. Un *joystick* es más barato de producir y también se estropea menos, normalmente, pues está compuesto únicamente por contactos que se abren o cierran.

Por el contrario, un *paddle* está constituido por un potenciómetro, componente que lleva una pieza móvil, llamada cursor, que podemos desplazar a lo largo de una pista resistiva, dispuesta en sentido circular. El principal fallo que puede ocurrir es el desgaste de la pista, por el uso, y que los valores obtenidos no correspondan a la resistencia que debería presentar en ese punto, pero un potenciómetro tampoco es algo que se gaste todos los días. De hecho los utilizamos todos los días en nuestra radio y duran su tiempo. Es el mismo componente lo que se utiliza en el *paddle* que en una radio o tocadiscos, aunque la principal diferencia estriba en que el *paddle* requiere que el potenciómetro sea lineal, es decir, su resistencia será directamente proporcional al ángulo girado por el eje. En un sistema de audio suele emplearse el potenciómetro de característica logarítmica, o lo que es igual, la resistencia es proporcional al logaritmo del ángulo girado. Este factor es algo a tener en cuenta, si bien la apariencia externa del potenciómetro es la misma, el efecto conseguido no es igual. Veamos lo que sucedería en caso de utilizar uno logarítmico, por ejemplo, en el programa de demostración que ofrecemos más adelante. En un extremo (o arriba o abajo) de la pantalla, la nave espacial se desplazaría un largo trecho con un giro mínimo del eje del potenciómetro, sin embargo apenas se movería con la misma cantidad de giro en el extremo opuesto. Este efecto podría corregirse aplicando las funciones de carácter logarítmico del BASIC a un programa adecuado, pero son ganas de complicar las cosas sin necesidad.

Llegados a este punto no está de más reiterar que los *paddles* utilizados para los equipos de **Atari** son perfectamente válidos para el **Vic-2Ø** y el **C-64**, siendo compatible hasta el conector que los une al ordenador.

El montaje que presentamos se compone de dos posibilidades. La primera son dos *paddles* independientes entre sí, aunque comparten el mismo conector. La segunda son los dos *paddles* dispuestos a modo de *joystick*. Se puede utilizar uno u otro según sea el programa con que lo utilicemos, si por ejemplo sólo necesitamos movernos en una dirección, mientras nuestro contrincante lo hace en la paralela, o contrariamente, el

```
100 REM ***************
110 REM * PADDLES VIC *
120 REM ***************
130 REM
1000 REM *MEMORIA DE PANTALLA*
1010 MP=7680:MC=38400
2000 REM *INICIALIZACION DDR*
2010 POKE 37139,0:POKE37154,127
3000 REM *LECTURA PADDLES*
3010 X=PEEK(36872):Y=PEEK(36873)
4000 REM *COORDENADAS CARACTER*
4010 XO=INT((X/255)*21):YO=INT((Y/255)*23)
5000 REM *IMPRIME CARACTER*
5010 POKE MP+XO+22*YO,81:POKE MC+XO+22*YO,2
5020 POKE MP+XO+22*YO,81:POKE MC+XO+22*YO,1
5030 GOTO 3000
```

Figura 3. Programa de comprobación para el **Vic-2Ø**.

tu Paddle

control de movimientos debe ser total por medio del bastoncillo.

En el primer caso, es necesario que dispongamos de dos potenciómetros, por separado, en sendas cajas, siendo el diagrama de patillas el mostrado en la figura 2 (ver el diagrama del conector en la figura 5). La foto 1 muestra un posible aspecto externo del montaje de uno de los *paddles*, con el botón de fuego correspondiente (se utiliza igual que con los *joysticks*, tanto en el programa como en la conexión al ordenador).

La foto 2 muestra la sencillez de la conexión de los cables por dentro, y la 3 el aspecto que ofrece el conector

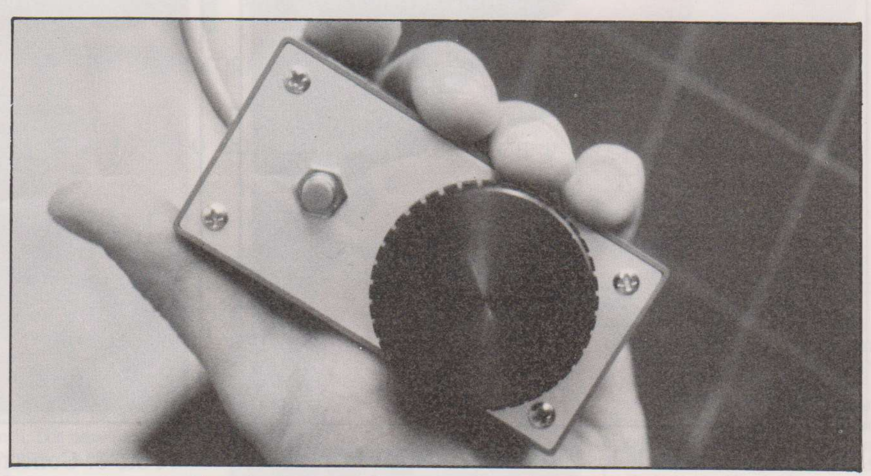


Foto 1. Aspecto de un *paddle* de mando único.

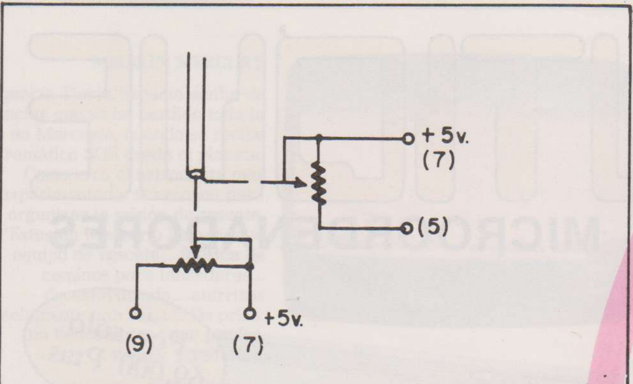
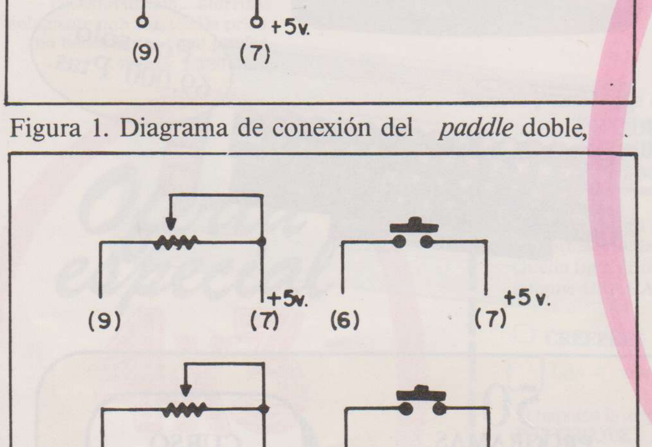


Figura 1. Diagrama de conexión del *paddle* doble.



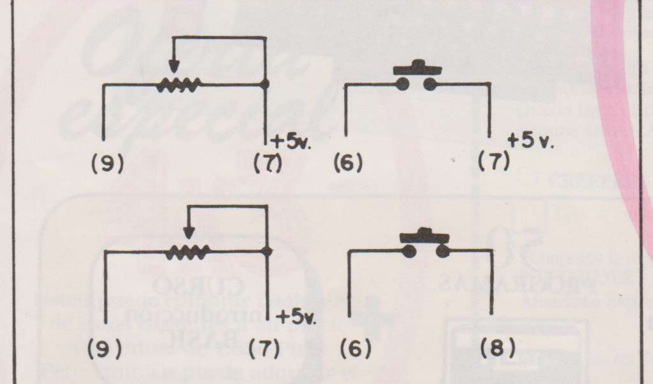


Figura 2. Diagrama de conexión de los dos *paddles* independientes.

tipo D de 9 contactos, que utilizamos para conectar este accesorio al **Vic-2Ø** o el **C-64.**

Si optamos por el doble *paddle*, tipo *joystick*, debemos localizar en el comercio el montaje mecánico adecuado. Es el mismo tipo que los empleados en los sistemas de radiocontrol. Varios son los aspectos que pueden ofrecer; en la foto 4 y en la 5 ofrecemos dos versiones distintas, aunque en realidad es el mismo componente.

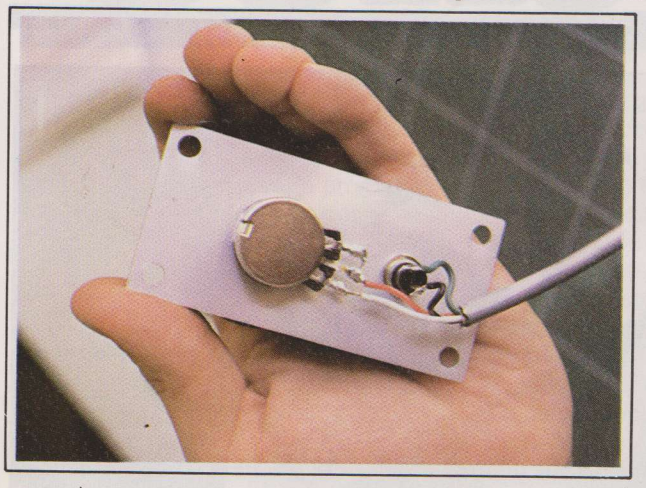


Foto 2. Conexión de los cables al potenciómetro.

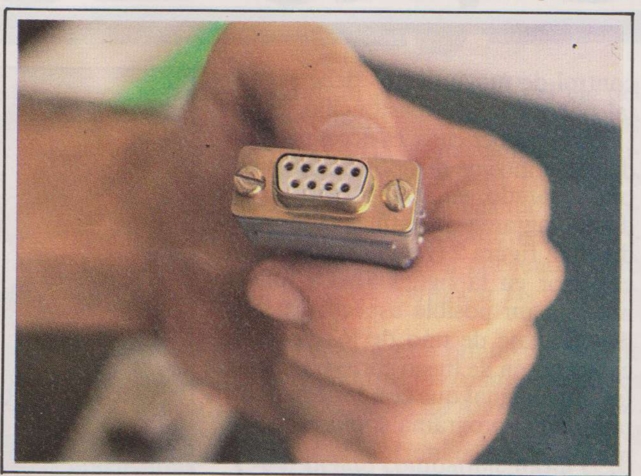


Foto 3. Conector tipo D de 9 patillas.

Tal vez sea difícil localizar uno que tenga los potenciómetros del valor que necesitamos, **47Ø Kohmios** en todos los casos. En tal circunstancia lo apropiado será buscar un modelo que nos permita cambiar los potenciómetros del valor original por el que se adecua a nuestras necesidades. Si el valor del potenciómetro fuera menor, no alcanzaremos a los valores superiores del rango posible, aunque hagamos el giro completo. Por el contrario, si el potenciómetro es de mayor valor, recorreremos toda la gama de valores posibles en la lectura del *paddle* antes de alcanzar el límite del giro. De todas maneras, como el potenciómetro no llegaría a girar los 270° totales, el valor deberá ser mayor.

Una vez realizado el montaje pertinente, siguiendo las fotos y figuras del artículo, solamente nos queda ver cómo funciona. Recordemos una vez más que el mismo montaje es válido tanto para el **Vic-2Ø** como para el **64**. También conviene hacer hincapié en que el diagrama del conector del *port* de control muestra la numeración de sus patillas viéndolo de frente, por tanto, debemos tenerlo en cuenta para hacer que cada cable del *paddle* vaya al lugar adecuado, de lo contrario, lógicamente, no funcionará.

En el **Vic-2Ø** es utilizado *chip* VIA 6522 para leer el valor de los potenciómetros. En la dirección 36872 se almacena el valor del potenciómetro X y la 36873 la del potenciómetro Y.

```
100 REM ****************
110 REM *  PADDLES 64   *
120 REM ****************
130 REM
1000 REM *CARGA DEL SPRITE*
1010 FOR NUM=0 TO 63
1020 READ SPR
1030 POKE(13*64)+NUM,SPR
1040 NEXT NUM
1050 POKE 2040,13
1060 REM *DATAS DEL SPRITE*
1070 DATA0,0,0,0,0,0
1080 DATA1,0,128,3,144,128
1090 DATA3,254,128,3,255,128
1095 DATA 3,255,128,7,255,192
1100 DATA47,255,232,63,255,248
1110 DATA47,255,232,0,254,0
1120 DATA7,255,192,3,255,128
1130 DATA1,255,0,7,147,192
1140 DATA12,16,96,8,16,32
1150 DATA8,16,32,28,56,112
1160 DATA0,0,0,0
1200 REM *SELECCION DEL COLOR*
1210 POKE53287,1
1300 REM *ACTIVAR SPRITE*
1310 POKE53269,PEEK(53269)OR1
1400 REM *LECTURA DE LA COORDENADA X*
1410 XO=2*PEEK(54298):POKE53248,(XOAND255):
1420 POKE53264,PEEK(56234)AND(255+(XO<256))OR ABS(XO>255)
1500 REM *LECTURA DE LA COORDENADA Y*
1510 YO=PEEK(54297):POKE53249,YO
1520 FORA=1 TO 50:NEXTA
1530 GOTO1400
```

Figura 4. Programa para el C-64.

Obteniendo en ellos valores comprendidos entre Ø y 255, según la posición del *paddle*. Pero también es necesario establecer los valores adecuados en el DDR, registro de dirección de los datos, que controla si los datos que nos interesan entran o salen del ordenador.

Así, ésta breve rutina nos dará una idea del funcionamiento:

```
1Ø POKE 37139, Ø: REM DDR PORT A
2Ø POKE 37154, 127: REM DDR PORT B
3Ø PRINT PEEK (36872); " "; PEEK (36873)
4Ø GOTO 2Ø
```

visualizando los consecutivos valores que van proporcionando los *paddles* en su movimiento.

Los botones de fuego se leen de la misma manera que vimos en la primera parte con los *joysticks*.

En el caso del **C-64**, los *paddles* son leídos por medio del *chip* de sonido SID, utilizándose direcciones de me-

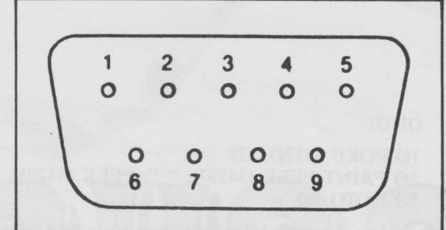


Figura 5. Numeración de los terminales del *port* de control, visto de frente.

PATILLA	VIC 20	CBM-64	
	PORT	PORT 1	PORT 2
1	JOY Ø	JOY AØ	JOY BØ
2	JOY 1	JOY A1	JOY B1
3	JOY 2	JOY A2	JOY B2
4	JOY 3	JOY A3	JOY B3
5	POT Y	POT AY	POT BY
6	LAPIZ OPTICO	BOTON A/L.O.	BOTON B
7	+5V	+5V	+5V
8	MASA	MASA	MASA
9	POT X	POT AX	POT BX

moria que corresponden a registros del mismo. Aquí, la dirección de memoria 54297 se encarga de leer el potenciómetro X, mientras que la 54298 lo hace con el potenciómetro Y. Esta breve rutina leerá sucesivamente los valores proporcionados por uno y otro:

```
1Ø POKE 5632Ø, 128
2Ø PRINT PEEK (54297); " "; PEEK (54298)
3Ø GOTO 2Ø
```

y visualizará valores entre Ø y 255 en ambas columnas. Para detener la ejecución habrá que recurrir a las teclas Run/Stop y Restore simultáneamente.

Los dos cortos programas de las figuras muestran en la 3 una demostración de posibilidades con el **Vic-2Ø** y en la 4 con el **C-64.** En futuros artículos se presentarán nuevos programas adecuados a *joysticks* y *paddles*.

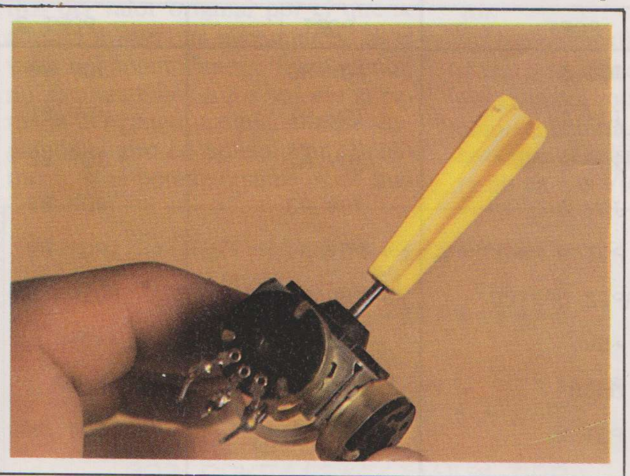


Foto 4. *Paddle* doble, un modelo comercializado.

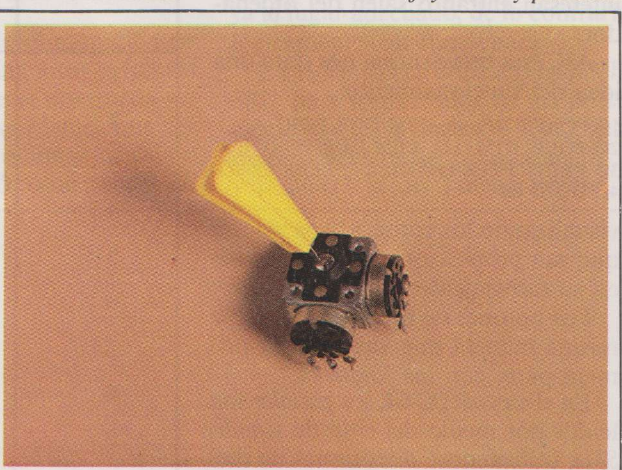
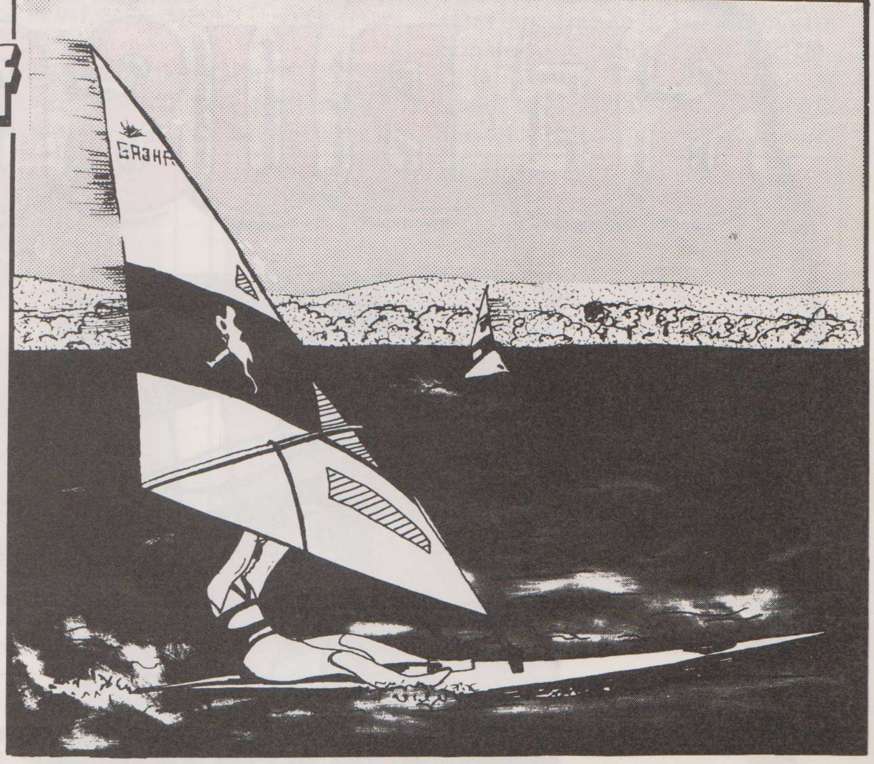


Foto 5. Otro modelo de *paddle* doble y suconexión

Concurso

Windsurf



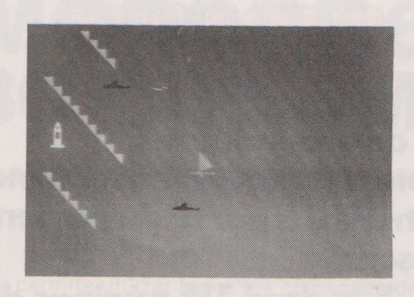
Ahora que llega el frío se nos ha ocurrido que puede ser muy agradable jugar con este programa llamado **WINDSURF**, que nos remite **Ignacio Ortega Ortigosa** desde Algeciras, y recordar los buenos días de playa de este pasado verano durante los cuales, más de uno se habrá pasado horas luchando contra las olas con su tabla de *windsurf*.

El programa es la historia de un intrépido *windsurfer* que acude en socorro del torrero de un faro. Después de varios días de furioso temporal, el torrero se ha quedado sin provisiones, y tú, el «*windsurfer*», has decidido acudir en su ayuda por lo que vas a intentar llegar hasta el faro en tu vela que manejarás con un *joystick* conectado al *port* 2.

Dos peligros se ciernen sobre ti: el primero de ellos es el fuerte oleaje de la zona que hará que, si eres alcanzado por la rompiente, tengas que comenzar de nuevo la travesía. El segundo es un poco más siniestro pues resulta que la zona está llenita de tiburones con hambre de varios días, que te atacarán cuando menos te lo esperes.

El faro lanza destellos de luz roja que te servirán de guía y te avisarán del peligro.

Tu tabla es fuerte, pero si te caes muchas veces acabará por romperse con lo que quedarás eliminado y a merced de los tiburones.



Las líneas que siguen son una descripción de la estructura del programa:

9	—	1Ø • 5Ø	Pantalla y constantes
55	—		Apaga destellos
6Ø	—	12Ø	Rutina oleaje
13Ø	—		Anulación *Sprites*
14Ø	—	17Ø	Finales varios
21Ø	—	32Ø	*Joystick* y colisiones
5ØØ	—		Tabla a su origen y restaura colisión
51Ø	—	59Ø	Rutina final
6ØØ	—	695	Activa sprites
7ØØ	—	79Ø	Niveles de juego
1ØØØ	—.	1Ø14	Información digital dibujos

CBM 64

```
0 REM **********************************************
1 REM *                                            *
2 REM *              WINDSURF POR ORTEGA # 1984    *
3 REM *                                            *
4 REM **********************************************
9 POKE53280,6:POKE53281,6:PRINT"▒"
10 TX=220:TY=180:V=53248:GOSUB700:GOSUB600
50 PRINT"▒":CT=6:TI$="000000"
55 POKE(V+21),PEEK(V+21)AND(255-2↑3)
58 TX=INT(150*RND(1))+80:TY=INT(150*RND(1))+80
60 FORL=0TOCT
```

```
65 GOSUB210
70 PRINT"█";
80 FORI=1TOVE:NEXTI,L
85 CA=NT(6*RND(1))+1
90 FORL=1TOCA:POKE(V+21),PEEK(V+21)OR(2↑3):PRINT'
95 IFCT>20THENPRINT"██████████                    ":PRINT"
100 CT=CT+1:CM=CT*10:IFCT=23THENCT=6
110 NEXT
120 GOTO55
130 FORL=1TO5:POKE(V+21),PEEK(V+21)AND(255-2↑L):NEXT:RETURN
140 GOSUB130:PRINT"J"TAB(7)"█████POR"SC"CHAFUZONES.TTTTT":RETURN
150 GOSUB130:PRINT"J"TAB(7)"██████ LO  CONSEGUISTE .TTTTT":RETURN
160 GOSUB130:PRINT"J"TAB(7)"██████ !! MENUDO MORDISCO !! "
170 PRINTTAB(6)"█ESTA CLARO QUE HAS PERDIDO.TTTTTTT":RETURN
210 A=PEEK(56320):A=(NOT(A))AND31
215 IFPEEK(V+31)=2THENX=240:Y=50:SC=SC+1:IFSC>CSTHENGOSUB140:GOTO500
216 IFPEEK(V+30)=6THENGOSUB150:GOTO500
217 IFPEEK(V+30)=18THENGOSUB160:GOTO500
218 IFPEEK(V+30)=34THENGOSUB160:GOTO500
220 IFAAND1THENY=Y-VA:TX=TX-4:IFTX<50THENTX=50
222 CM=CM-2:IFCM<60THENCM=220
230 IFY<50THENY=50
240 IFAAND2THENY=Y+VA:TX=TX-4:IFTX<50THENTX=50
250 IFY>231THENY=231
260 IFAAND4THENX=X-VA:TX=TX-4:IFTX<50THENTX=50
270 IFX<20THENX=20
280 IFAAND8THENX=X+VA:TX=TX-4:IFTX<50THENTX=50
290 IFX>240THENX=240
300 POKEV+2,X:POKEV+8,TX:POKEV+10,CM
310 POKEV+3,Y:POKEV+9,TY:POKEV+11,CM
320 RETURN
500 POKEV+2,240:POKEV+3,50:POKE(V+30),0
510 PRINT"████████████████████████JUEGO TERMINADO EN"
520 PRINT"█           "MID$(TI$,3,2);" MINUTOS
530 PRINT"█           "RIGHT$(TI$,2);" SEGUNGOS
550 PRINT"█        █ OTRO JUEGO ?  (S/N) █"
560 GETW$:IFW$<>"S"ANDW$<>"N"THEN560
570 IFW$="S"THENGOSUB130:RUN10
580 IFW$="N"THENGOSUB130:PRINT"████████████ENTONCES...."
590 PRINT"██████PON A SECAR LA VELA████████████████":END
600 X=240:Y=50
610 IFMA=1THENPOKEV+21,30
615 IFMA=2ORMA=3THENPOKEV+21,62
620 POKE2041,13:POKE2042,14:POKE2043,15:POKE2044,11:POKE2045,11
621 :
622 :
623 :
624 :
625 :
630 FORS=832TO894:READQ:POKES,Q:NEXT
640 FORS=896TO958:READW:POKES,W:NEXT
650 FORS=960TO1022:READQ1:POKES,Q1:NEXT
655 FORS=704TO766:READQ2:POKES,Q2:NEXT
660 POKEV+2,X:POKEV+3,Y
670 POKEV+4,25:POKEV+5,128
680 POKEV+6,45:POKEV+7,120
683 POKEV+8,TX:POKEV+9,TX:POKEV+10,TY:POKEV+11,TY
685 POKEV+29,8
690 POKEV+40,4:POKEV+41,7:POKEV+42,10:POKEV+43,0:POKEV+44,0
695 RETURN
700 PRINTTAB(17)"█████GRADO DE DIFICULTAD"
710 PRINTTAB(21)"██1    MAREJADA"
```

PROGRAMA ★ commodore Magazine ★ COMPROBADO

PREMIADO CON 5.000 PESETAS

Viene de la página anterior

```
720 PRINTTAB(21)"██2   FUERTE MAREJADA"
730 PRINTTAB(21)"██3   MAR GRUESA"
740 PRINTTAB(21)"██";:INPUT"CUAL ELIGES";MA
750 IFMA<1ORMA>3THEN700
760 IFMA=1THENVE=100:VA=8:CS=5
770 IFMA=2THENVE=50:VA=7:CS=7
780 IFMA=3THENVE=2:VA=6:CS=9
790 RETURN
1000 DATA0,64,0,0,96,0,0,112,0,0,120,0,0,124,0,0,126,0,0,127,0,0,127,128
1001 DATA0,127,192,0,127,224,0,127,240,0,255,252,0,127,240,0,127,128,0,124,0
1002 DATA0,96,0,0,64,0,127,255,254,0,112,4,0,48,0,0,16,0
1005 DATA0,24,0,0,24,0,0,60,0,0,36,0,0,36,0,0,102,0,0,126,0,0,126,0
1006 DATA0,66,0,0,66,0,0,66,0,0,66,0,0,66,0,0,66,0
1007 DATA0,126,0,0,126,0,0,126,0,0,126,0,0,126,0
1008 DATA3,255,192,3,255,192
1010 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1012 DATA0,7,255,15,255,255,255,255,255,1,255,255,0,1,255
1014 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1020 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1022 DATA0,0,0,0,0,0,0,0,0,0,56,0,0,112,0,0,240,0,15,248,0
1024 DATA111,255,193,255,255,255
```

Minas ocultas

VIC-20

MINAS OCULTAS es un juego para el **VIC 2Ø** que ha enviado a concurso **Carlos Plumed**. El juego consiste en desplazarse por la pantalla mediante las teclas I, J, K, M, con el fin de recoger 15 «corazones» que se encuentran distribuidos aleatoriamente. En cada una de las diversas pantallas en que se desarrolla el juego hay un cierto número de minas ocultas, número que depende de la pantalla y del nivel de juego elegido (1 = fácil, 3 = difícil). Cuando se tenga la mala suerte de tropezar con una mina se perderá una de vida de las tres de que se dispone inicialmente.

Para la recogida de los corazones se dispone de dos «artilugios» o inventos que se accionan mediante las teclas de función F1 y F7. F1 controla algo así como un «aparato para saltar» que permite realizar saltos largos y aleatorios y que puede utilizarse para acercarse a los corazones más rápidamente.

Si al dar el salto se cae encima de una mina pues «bum», se pierde una vida. Además durante el salto se deja una traza o rastro en la pantalla que si se vuelve a tocar restará puntos.

La tecla F7 sirve para accionar el «desactivador automático de minas». Su accionamiento elimina cualquier mina que hubiera en las casillas adyacentes a la del jugador y aumenta la puntuación. Sólo se puede accionar cinco veces por cada pantalla y su empleo llena de asteriscos todas las casillas adyacentes a la del jugador.

La recogida de corazones debe hacerse en un minuto por lo que es preciso actuar con rapidez y contar con buena suerte.

Para los que quieran meterse en modificaciones y mejoras adjuntamos unas líneas describiendo la estructura del programa:

12	—	Datos para localizar las ocho casillas adyacentes
2Ø	—	Inicialización vidas, antiminas y comienzo de la primera pantalla
22	—	Salto a subrutina instrucciones
13Ø	— 16Ø	Sitúa espacios en blanco (de 13 a 19 según nivel)
17Ø	— 172	Colocan 15 minas aleatoriamente
173	—	Colocación del jugador
174	— 179	Comprueba vidas y tiempo
19Ø	— 195	Movimiento elegido, F1, F7, I, J, K, M,
2ØØ	—	Resultado del movimiento
22Ø	—	Sitúa un círculo en las posiciones ya ocupadas
2999	— 32ØØ	Fin de juego

Subrutinas:

5ØØ	—	Antiminas
2ØØØ	—	Cambio de pantalla y reinicio del juego
5ØØØ	—	Instrucciones y elección de nivel
7ØØØ	—	Sonido
9ØØØ	—	Explora y dibuja sobre la pantalla

PREMIADO CON 5.000 PESETAS

PROGRAMA COMPROBADO

```
0 REM******************
1 REM*                *
2 REM* MINAS OCULTAS*
3 REM*                *
4 REM*       DE       *
5 REM*                *
6 REM*     CARLOS     *
7 REM*                *
8 REM*     PLUMED     *
9 REM******************
12 DATA-22,-21,1,23,22,21,-1,-23
13 FORN=1TO8:READX(N):NEXT
20 PRINT"▯":POKE36879,29:POKE36878,15:ZQ=10:V=3:M=5:HT=1
22 GOSUB5000
25 PRINT"▯▮▮▮▮▮▮▮▮▮▮▮▮▮PANTALLA="HT:FORN=1TO2500:NEXT
100 PRINT"▯"
130 FORN=1TOZQ
140 P=INT(RND(1)*484)+7702:Y=30720+P
141 IFPEEK(P)=96THEN140
150 POKEP,96
160 NEXT
170 FORN=1TO15
171 A=INT(RND(1)*484)+7702:K=A+30720:IFPEEK(A)=96 OR PEEK(A)=65 THEN 171
172 POKEA,65:POKEK,2:NEXT
173 G=INT(RND(1)*484)+7702:J=G+30720:IF PEEK(G)=96ORPEEK(G)=65THEN173
174 POKEG,160:POKEJ,6:TI$="000000"
176 H=0:S=59-VAL(TI$)
177 IF S<10THEN L=1
179 PRINT"▮▮VID="V"M="M"TIME="SPC(L)S:IFV<1ORS<1THEN2999
190 GETA$:IFA$="▮"ANDM>0ANDG<8163ANDG>7724THENGOSUB500
191 IFA$="■" THEN H=INT(RND(1)*484)+7702-G:GOTO 200
192 IFA$="I"THENH=-22:GOTO200
193 IFA$="K"THENH=1:GOTO200
194 IF A$="M" THENH=22:GOTO 200
195 IF A$="J" THEN H=-1:GOTO 200
196 GOTO 176
200 IFG+H>8185ORG+H<7702THENGOTO176
202 IFPEEK(G+H)=96THENPOKEJ+H,0:POKEG+H,86:POKE36877,230:FORHJ=1TO1500:NEXT
203 IFPEEK(G+H)=86THENV=V-1:F=F-100:POKE36877,0:GOTO220
205 IFPEEK(G+H)=32THENPOKE36874,230:F=F+10:POKE36874,0:GOTO220
207 IFPEEK(G+H)=87THENPOKE36875,230:F=F-25:POKE36875,0:GOTO220
209 IFPEEK(G+H)=65THENPOKE36876,230:F=F+100:POKE36876,0:SD=SD+1:IFSD=15THEN 2000
220 POKEJ,0:POKEJ+H,6:POKEG,87:POKEG+H,160:G=G+H:J=J+H:GOTO176
500 POKE36875,128:FORGH=1TO500:NEXT:POKE36875,0
510 FORZ=1TO8
520 IFPEEK(G+X(Z))=65THEN530
522 IFPEEK(G+X(Z))=96THENPOKEJ+X(Z),0:POKEG+X(Z),86:GOSUB1000:POKEJ+X(Z),4:F=F+5
0
525 POKEJ+X(Z),2:POKEG+X(Z),42
530 NEXT
540 M=M-1:RETURN
1000 FORN=255TO128STEP-1:POKE36875,0:POKE36875,N:NEXT:POKE36875,0:RETURN
2000 HT=HT+1:L=0:GOSUB9000:PRINT"▯▮▮▮▮PUNTOS="F:ZQ=ZQ+RT:SD=0:M=5:V=3:GOSUB7000
2100 FORN=1TO500:NEXT:GOTO25
2999 GOSUB9000
3000 IFS<1THENPRINT"▯▮▮▮▮SE TE ACABO EL TIEMPO"
3100 IFV<1THENPRINT"▯▮▮▮▮ SE TE ACABO LA VIDA"
3150 PRINT"▮▮▮▮▮▮▮▮▮▮▮PUNTOS="F
3200 HT=1:L=0:ZQ=10:F=0:V=3:M=5:SD=0:FORN=1TO2500:NEXT:PRINT"▯":GOSUB6000:GOTO25
5000 GOSUB 7000
5100 PRINT"▮▮*** MINAS OCULTAS ***"
5101 PRINT:PRINT"TIENES QUE COGER 15 ♠"
5102 PRINT"PUEDES CHOCARTE CON MINAS QUE TE HARAN PERDER VIDAS"
```

Viene de la página anterior

```
5103 PRINT"DISPONES DE:1 MINUTO,3 VIDAS Y 5 ANTIMINAS POR PANTALLA"
5104 PRINT
5105 PRINT"    EL ANTI MINAS BUSCA MINAS A TU ALREDEDOR"
5106 PRINT
5107 PRINT"    PUEDES COLOCARTE EN OTRO SITIO DE LA PAN-TALLA APRETANDO F1"
5110 PRINT:PRINT"HAS DE MOVER:I-ARRIBA:K-DERECHA:M-ABAJO:J-IZQUIERDA:F1-HIPER ES
PACIO:";
5111 PRINT"F7-ANTI MINAS"
5120 PRINT"PULSA PARA EMPEZAR"
5199 GETFR$:IFFR$=""THEN5199
5200 GOSUB6000
5250 RETURN
6000 PRINT"[illegible]ELIGE NIVEL(1-3)"
6010 PRINT"[illegible]  A MAYOR NIVEL MAYOR     DIFICULTAD"
6020 GETAG$:IFAG$=""THEN6020
6050 RT=VAL(AG$):IF RT>3ORRT<1THEN6020
6055 RT=RT*3
6060 RETURN
7000 FORN=255TO128STEP-.5:POKE36875,0:POKE36875,N:NEXT:POKE36875,0:RETURN
9000 FOR N=7702TO8185
9010 IFPEEK(N)=96THENPOKEN,86:POKEN+30720,0
9020 NEXT
9030 FORM=1TO7000:NEXT
9040 RETURN
```

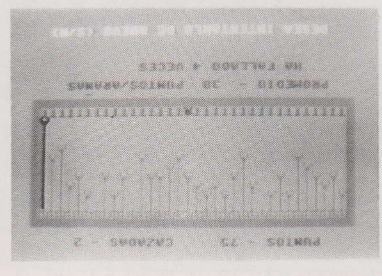
Concurso

Araña

Araña nos lo envía **Juan Luis Ruiz**, desde Burgos. Se trata de un juego para el **C-64** en el que el jugador tiene que hacer de cazador de arañas. Las arañas van descolgándose mediante un hilillo de tela de araña desde el techo de la habitación. El jugador, que puede desplazarse por el suelo de la habitación de izquierda a derecha, dispone de una pistola de matar arañas con la que debe evitar que ninguna araña llegue al suelo. El número de disparos, que se realizan con la barra de espacios, es ilimitado, de este modo, el juego consiste en tener a las arañas a raya el mayor tiempo posible, anticipándose a sus movimientos. Las instrucciones completas para jugar van incluidas en el propio programa.

El programa está estructurado de la siguiente manera:

21Ø — 51Ø	Mensaje de comienzo del juego
61Ø — 131Ø	Rutina de caracteres programables
141Ø — 164Ø	Inicializa sonido y primer efecto sonoro
165Ø — 2Ø1Ø	Mensaje de las teclas a pulsar
21ØØ — 271Ø	Rutina de dibujo de pantalla
281Ø — 282Ø	Dimensiona matriz, inicializa variables y activa modo
3Ø1Ø — 331Ø	Analiza la entrada de información
34ØØ — 372Ø	Efecto sonoro y desplazamiento del rayo
381Ø — 4Ø1Ø	Rutina de puntuación y arañas cazadas
411Ø — 421Ø	Desplazamiento del disparador laser
431Ø — 481Ø	Movimiento aleatorio de las arañas
49ØØ — 531Ø	Efecto sonoro, evaluación y mensaje fin de juego



CBM 64

```
10 REM  ******************
20 REM  **              **
30 REM  **       -      **
40 REM  **  A R A N A S **
50 REM  **              **
60 REM  ******************
70 REM
80 REM
110 GOSUB1710:POKE53280,6:POKE53281,6
210 PRINTLEFT$(A$,7)"■■■■ESTE PROGRAMA DE LAS '■ARAN■■■AS■'"
310 PRINT"■■TARDARA 15 SG. EN ESTAR PREPARADO."
410 PRINT"■■HASTA ENTONCES, RELAJESE, UN SONIDO LE"
510 PRINT"■■AVISARA CUANDO ESTE LISTO EL JUEGO."
610 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
710 FORI=0TO63:FORJ=0TO7
810 POKE12288+I*8+J,PEEK(53248+I*8+J)
910 NEXTJ,I
1010 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
1110 POKE53272,(PEEK(53272)AND240)+12
1210 FORCA=58TO63:FORBY=0TO7
1310 READNU:POKE12288+(8*CA)+BY,NU:NEXTBY,CA
1410 GOSUB1720
1510 POKE54296,15:POKE54277,64:POKE54278,64:POKE54276,17
1610 POKE54273,34:POKE54272,75:FORT=1TO250:NEXT
1615 POKE54273,51:POKE54272,97:FORT=1TO450:NEXT
1620 POKE54273,34:POKE54272,75:FORT=1TO300:NEXT
1625 FORI=80TO140STEP2.5:POKE54273,I:NEXT
1630 FORI=200TO50STEP-3:POKE54273,I:NEXT
1640 POKE54273,0:POKE54296,0:POKE54276,0
1650 GOTO1750
1710 A$="■■■■■■■■■■■■■■■■■■■■■■■■■":PRINT"■":RETURN
1720 FORL=54272TO54296:POKEL,0:NEXT:RETURN
1750 GOSUB1710:PRINTLEFT$(A$,10)SPC(10)"■LOS MANDOS SON"
1760 PRINTSPC(6)"■■'■Z■' - IZQUIERDA   '■X■' - DERECHA"
1770 PRINTSPC(6)"■■■BARRA ESPACIADORA■ - RAYO LASER"
1810 PRINTSPC(6)"■■■■PULSE '■S■' SI ESTA PREPARADO."
```

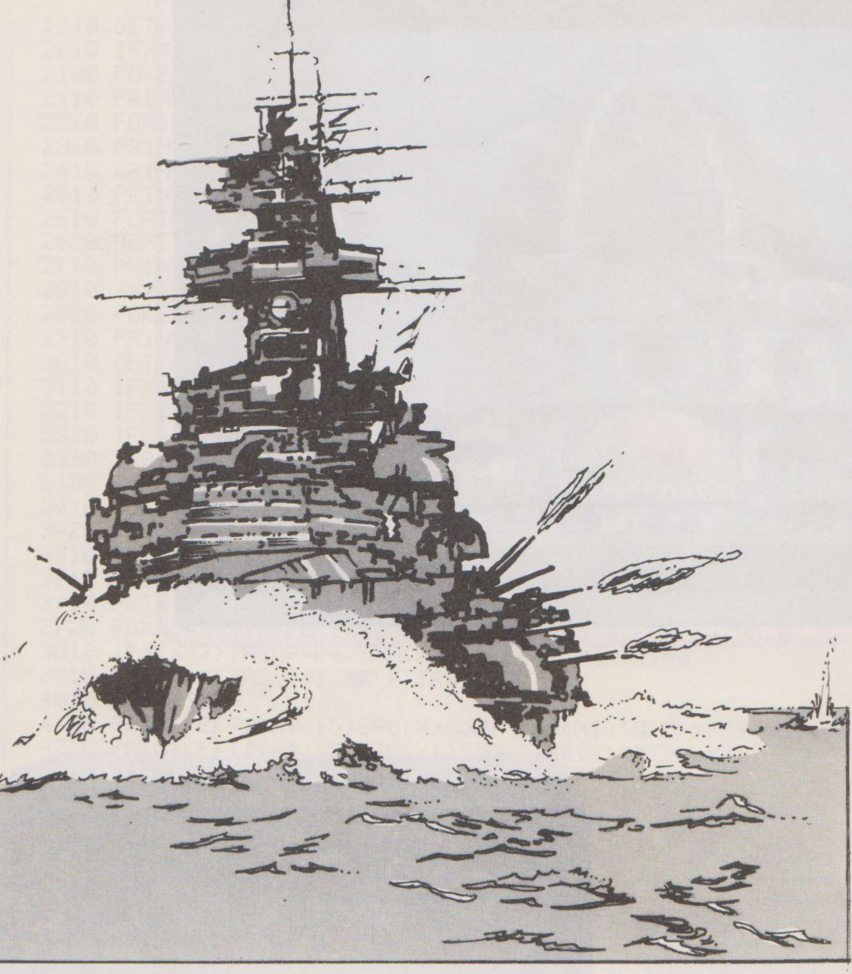
PREMIADO CON 5.000 PESETAS

Concurso

```
1910 GETX$:IFX$="S"THEN2100
2010 IFX$<>"S"THEN1910
2100 POKE53280,8:POKE53281,8:PRINT"█"
2110 PRINTLEFT$(A$,3)SPC(2)"█>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>":REM 36 - >
2210 FORI=4TO15:PRINTLEFT$(A$,I)SPC(2)">"SPC(34)">";:NEXT
2310 PRINTLEFT$(A$,16)SPC(2)">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>":REM 3 - >
2410 PRINT"█"TAB(6)"██PUNTOS -"P;TAB(22)"CAZADAS -"C
2510 PRINTLEFT$(A$,4)SPC(3)"██??????????????????????????????????";:REM 34 - ?
2610 FORI=5TO14:PRINTLEFT$(A$,I)SPC(3)"██
2620 NEXT
2710 PRINTLEFT$(A$,15)SPC(3)"██TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT";:REM 34 - T
2810 X=20:DIMA(34):P=0:C=0
2820 POKE53265,PEEK(53265)OR64:POKE53283,4
2910 PRINTLEFT$(A$,15)SPC(X)"██":X1=X:GOSUB1720
3010 GETX$
3110 IFX$="Z"ANDX>3THENX1=X-1:GOTO4210
3210 IFX$="X"ANDX<36THENX1=X+1:GOTO4110
3310 IFX$<>" "THEN4010
3390 D=D+1
3400 POKE54296,12:POKE54280,1:POKE54279,116:POKE54284,15:POKE54283,129
3410 FORI=13TO5STEP-1
3420 POKE54296,(I-4)
3510 PRINTLEFT$(A$,I)SPC(X1)"██"
3610 PRINTLEFT$(A$,I+1)SPC(X1)"█ ":NEXT
3710 PRINTLEFT$(A$,5)SPC(X1)"█ "
3720 POKE54279,0:POKE54280,0:POKE54283,0:POKE54296,0
3810 IFA(X-2)THENP=P+25*A(X-2):C=C+1:A(X-2)=0
3910 PRINT"█"TAB(14)"██"P;TAB(31)C
4010 GOTO4310
4110 PRINTLEFT$(A$,15)SPC(X)"██T"TAB(X1)"█":X=X1:GOTO4310
4210 PRINTLEFT$(A$,15)SPC(X1)"██"TAB(X)"█T":X=X1
4310 IF(RND(1)*10000)>(P+8000)THEN3010
4410 A=INT(RND(1)*34)+1
4510 PRINTLEFT$(A$,A(A)+5)SPC(A+2)"██="
4610 PRINTLEFT$(A$,A(A)+6)SPC(A+2)"█K"
4710 A(A)=A(A)+1:IFA(A)=9THEN4900
4610 GOTO3010
4900 FORI=5TO13:PRINTLEFT$(A$,I)SPC(A+2)"██=":NEXT
4910 PRINTLEFT$(A$,A(A)+5)SPC(A+2)"█K"
4920 GOSUB1720:POKE54296,15:POKE54277,190:POKE54278,64:POKE54276,17
4930 FORI=1TO30STEP.2:POKE54273,I:NEXT
4935 FORI=50TO20STEP-.2:POKE54273,I:NEXT
4940 FORI=10TO40STEP.2:POKE54273,I:NEXT
4980 POKE54273,0:POKE54276,0:POKE54296,0
4990 PC=INT(P/C):IF(P/C-INT(P/C))>.4THENPC=INT(PC)+1
5000 DI=D-C:IFDI>0THENPA=1
5010 PRINTLEFT$(A$,18)"██       PROMEDIO - "PC" PUNTOS/ARAN█▀█AS"
5020 IFDI=1THENPRINTSPC(10)"█HA FALLADO UNA VEZ":GOTO5050
5030 IFPA=1THENPRINTSPC(8)"█HA FALLADO"DI"VECES":GOTO5050
5040 PRINTSPC(8)"█NO HA FALLADO NINGUNA VEZ."
5050 PRINTSPC(5)"████DESEA INTENTARLO DE NUEVO (S/N)":CLR
5110 GETX$:IFX$="S"THEN1750
5210 IFX$="N"THENSYS64759
5310 GOTO5110
6010 DATA24,255,126,126,126,60,231,195
6110 DATA24,56,96,124,30,6,28,24
6210 DATA8,24,60,60,90,129,129,66
6310 DATA8,8,8,8,8,8,8,8
6410 DATA255,255,255,255,255,255,255,255
6510 DATA136,235,156,136,136,207,60,8
```







Batalla naval

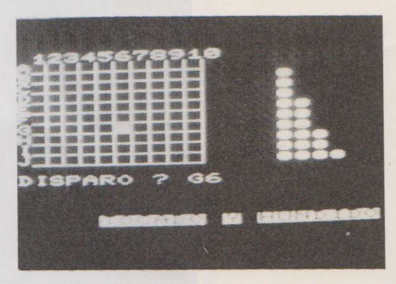
VIC 2Ø + EXPANSION DE 3K

José M. Pérez Murano, lector de **Commodore Magazine** en Barcelona, es el autor de «BATALLA NAVAL». El programa es una versión del popular juego de los «barquitos», pero en el que el contrincante va a ser tu ordenador **VIC** en lugar de cualquier otra persona.

El programa presenta en la pantalla una cuadrícula de 1Ø × 1Ø sobre la que se van efectuando los disparos por medio del teclado. Cada disparo efectuado deja su señal en la pantalla, tanto si ha sido un disparo al agua como si el impacto ha dado de lleno en algún barco. En cada uno de los casos el carácter sobre la pantalla y el sonido que puede escucharse son diferentes.

El número y dimensiones de los barcos son los reglamentarios de este juego, es decir: 1 barco de 4 casillas, 2 de 3, 3 de 2, y 4 de 1 (estos últimos son los submarinos).

En el margen derecho de la pantalla aparecen representados en todo



momento los barcos que quedan por destruir, desapareciendo la representación de cada barco cuando resulta hundido.

Al terminar la partida podremos ver el número de los disparos efectuados y tendremos la posibilidad de jugar de nuevo o de terminar con el programa.

La estructura del programa es la siguiente:

1ØØØ	11ØØ	Comprueba que no se tocan los barcos
2ØØØ	2Ø3Ø	Presentación
3ØØØ	3Ø2Ø	Crea barcos
2Ø	7Ø	Coloca el barco de 4
8Ø	25Ø	Coloca dos barcos de 3
3ØØ	47Ø	Coloca tres barcos de 2
49Ø	54Ø	Coloca cuatro barcos de 1
6ØØ	62Ø	Entrada disparo
7Ø9	71Ø	Pinta tablero
63Ø		Agua
635		Tocado hundido
6Ø1Ø		Sonido de agua
698	72Ø	Pinta tablero modificado
6ØØØ		Ruido tocado
635	653	Comprueba si es hundido
4ØØØ	4Ø9Ø	Fin. Dibuja tablero completo y acaba
655	675	Borra los barcos hundidos

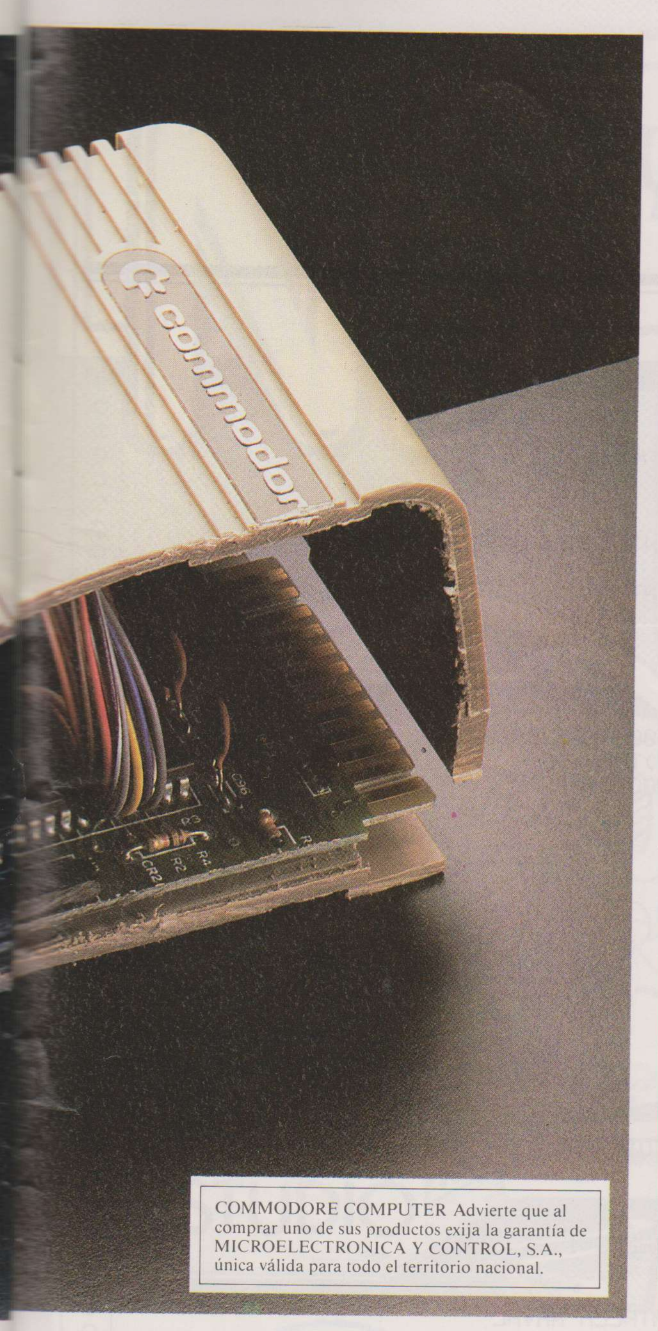
```
1 POKE36878,15
2 GOSUB2000
3 POKE36879,8:PRINT"♥"
4 GOSUB3000
10 DIMA(12,12)
20 X=INT(RND(1)*10)+1
30 Y=INT(RND(1)*10)+1
40 IF(X>7ANDY>7)THEN20
50 A(X,Y)=4
60 IFX>7THENA(X,Y+1)=4:A(X,Y+2)=4:A(X,Y+3)=4:GOTO80
70 A(X+1,Y)=4:A(X+2,Y)=4:A(X+3,Y)=4
80 FORT=1TO2
90 X=INT(RND(1)*10)+1
100 Y=INT(RND(1)*10)+1
110 IFX>8ANDY>8THEN90
115 XX=X:YY=Y

125 IF(X<8ANDY<8)THEN200
130 IFY>8THEN170
140 FORK=0TO2:YY=Y+K:GOSUB1000:NEXT:IFJ=1THENJ=0:GOTO90
145 IFH=3THENH=6:GOTO160
150 H=3
160 FORK=0TO2:A(X,Y+K)=H:NEXT:GOTO250
170 FORK=0TO2:XX=X+K:GOSUB1000:NEXTK:IFJ=1THENJ=0:GOTO90
172 IFH=3THENH=6:GOTO180
175 H=3
180 FORK=0TO2:A(X+K,Y)=H:NEXT:GOTO250
200 IFX/2=INT(X/2)THEN140
210 GOTO170
250 NEXT
300 FORT=1TO3
310 X=INT(RND(1)*10)+1
320 Y=INT(RND(1)*10)+1
330 IF(X>9ANDY>9)THEN310
340 XX=X:YY=Y

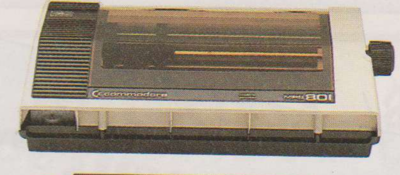
350 IF(X<9ANDY<9)THEN440
360 IFY>9THEN400
370 FORK=0TO1:YY=Y+K:GOSUB1000:NEXT
380 IFJ=1THENJ=0:GOTO310
382 IFH=5THENH=7:GOTO390
384 IFH=2THENH=5:GOTO390
386 H=2
390 FORK=0TO1:A(X,Y+K)=H:NEXT:GOTO470
400 IFY<9THEN440
410 FORK=0TO1:XX=X+K:GOSUB1000:NEXT
420 IFJ=1THENJ=0:GOTO310
422 IFH=5THENH=7:GOTO430
424 IFH=2THENH=5:GOTO430
426 H=2

430 A(X,Y)=H:A(X+1,Y)=H:GOTO470
440 IFX/2=INT(X/2)THEN410
450 GOTO370
470 NEXT
490 FORT=1TO4
500 X=INT(RND(1)*10)+1
510 Y=INT(RND(1)*10)+1
520 XX=X:YY=Y:GOSUB1000:IFJ=1THENJ=0:GOTO500
```

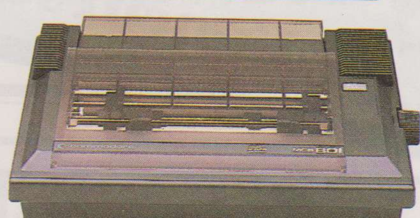
PROGRAMA commodore Magazine COMPROBADO



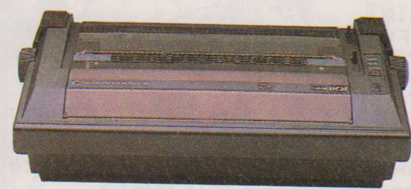
COMMODORE COMPUTER Advierte que al comprar uno de sus productos exija la garantía de MICROELECTRONICA Y CONTROL, S.A., única válida para todo el territorio nacional.



COMMODORE MPS 801
Impresora de matriz



COMMODORE MCS 801
Impresora cuatro colores



COMMODORE DPS 1101
Impresora de margarita

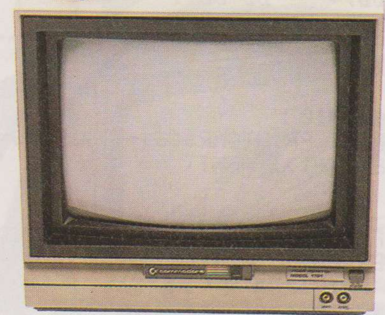


COMMODORE 1520
Printer plotter cuatro colores

COMMODORE 1541
Unidad de disco 170 K



COMMODORE 1530
Unidad de cassette



COMMODORE 1701
Monitor con color y sonido

JOYSTICKS
PADDLES
SOFTWARE Y LIBROS

10% de su cerebro?

lograr que los textos y gráficos sean más claros y vivos que en un televisor.

— 3 impresoras y un printer-plotter, para copias en papel, en blanco y negro o color.

— Los joystick y paddles, para conseguir que los juegos sean más excitantes.

Sea exigente. Si su cerebro, el de su COMMODORE 64, puede darle 100, no se conforme con 10.

commodore
COMPUTER
la emoción del futuro

Concurso

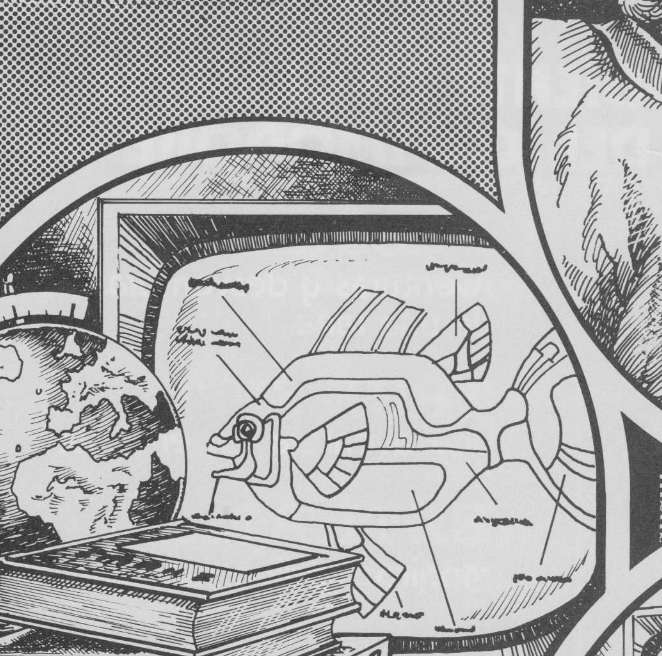
```
530 A(X,Y)=1
540 NEXT
590 PRINT"□":PRINT
595 FORX=1TO10:FORY=1 TO 10:T$(X,Y)="┘":NEXTY:NEXTX
596 GOTO709
600 PRINT:PRINT:AC=AC+1:INPUT"DISPARO ";A$
605 X=ASC(MID$(A$,1,1))-64:IFX<1ORX>10THENPRINT"TTT";:GOTO600
608 IFMID$(A$,2,1)=""THENPRINT"TTT";:GOTO600
610 Y=ASC(MID$(A$,2,1))-48:IFY>10ORY<1THENPRINT"TTT";:GOTO600
615 IFY=1ANDMID$(A$,3,1)="0"THENY=10
620 IF T$(X,Y)<>"┘"THENPRINT"TTT";:GOTO600
625 PRINT:PRINT
630 IFA(X,Y)=0THENPRINT"▒       ▒AGUA■                  ":T$(X,Y)="o":GOSUB6010:GOTO69
8
635 PRINT"▒      ▒TOCADO■";:T$(X,Y)="●":GOSUB5000:GOSUB6000
640 IFA(X,Y)<>1ANDA(X,Y)<5ANDQ(A(X,Y))<>A(X,Y)THEN698
645 IF(A(X,Y)=5ORA(X,Y)=7)ANDQ(A(X,Y))<>2THENGOTO698
650 IFA(X,Y)=6ANDQ(A(X,Y))<>3THENGOTO698
653 PRINT" ▒Y■ ▒HUNDIDO■"
655 W=A(X,Y)
656 IFW=5ORW=7THENW=2
657 IFW=6THENW=3
660 IFW=4THENE$(9)="    ":GOTO680
665 IFW=3THENE$(7+BB)="   ":BB=BB+1:GOTO680
670 IFW=2THENE$(4+CC)="  ":CC=CC+1:GOTO680
675 IFW=1THENE$(DD)=" ":DD=DD+1
680 HH=HH+1:IFHH=10THENGOTO4000
698 PRINT"                      "
699 PRINT"▒"
705 PRINT
709 PRINT" 12345678910□"
710 FORT=1TO10:PRINT:PRINTCHR$(64+T);:FORJ=1TO10:PRINTT$(T,J);:NEXTJ
715 PRINT"    ";E$(T-1);:NEXTT
720 GOTO600
999 END
1000 FORF=-1TO1:FORG=-1TO1:IFA(XX+F,YY+G)<>0THENJ=1
1010 NEXT:NEXT:RETURN
2000 POKE36879,8:PRINT"□▒"
2010 PRINT"▒▒▒***********************▒▒▒    BATALLA NAVAL      "
2020 PRINT"▒▒************************"
2030 RETURN
3000 FORT=0TO3:E$(T)="●":NEXT
3004 FORT=4TO6:E$(T)="●●":NEXT
3008 FORT=7TO8:E$(T)="●●●":NEXT
3010 E$(9)="●●●●"
3020 RETURN
4000 FORK=1TO7000:NEXT
4020 PRINT"□":PRINT" 12345678910";
4030 FORI=1TO10:PRINT:PRINTCHR$(64+I);:FORJ=1TO10
4040 IFT$(I,J)="o"THENT$(I,J)="┘"
4045 PRINTT$(I,J);:NEXTJ:NEXTI
4050 PRINT:PRINT:PRINT:PRINT:PRINT"HAS GANADO LA BATALLA"
4055 PRINT"▒DISPARANDO";AC;" VECES"
4060 PRINT:PRINT"DESEAS CONTINUAR (S/N)"
4070 GETA$:IFA$=""THEN4070
4080 IFA$="S"THENRUN
4090 END
5000 Q(A(X,Y))=Q(A(X,Y))+1:RETURN
6000 FORYY=1TO200:POKE36877,200:NEXT:POKE36877,0:RETURN
6010 FORYY=241TO120STEP-1:POKE36876,YY:NEXT:RETURN
```

PROGRAMA ★ commodore Magazine ★ COMPROBADO



SU TIENDA *amiga* DE INFORMATICA

GESTION



ESTUDIO

JUEGOS

COMMODORE – ORIC
AMSTRAD - SPECTRUM
SPECTRAVIDEO

LAS ULTIMAS NOVEDADES DE SONIMAG

¡¡YA DISPONIBLES!!

novo/digit
microinformatica

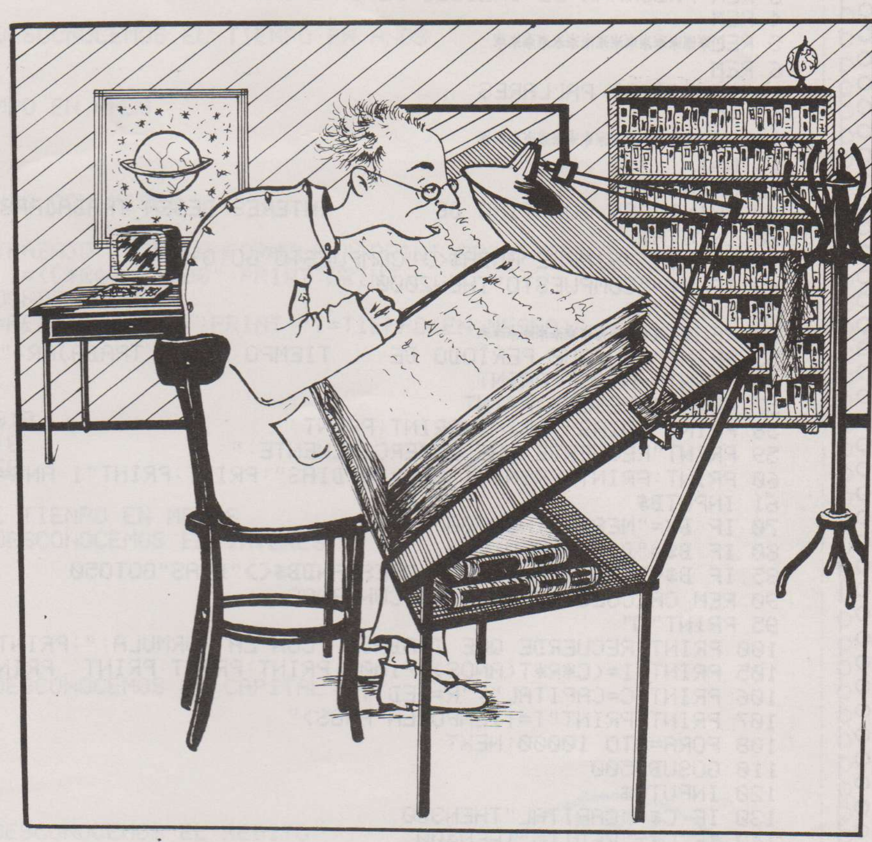
C/ Aragón, 472 - Telf. 246 27 75 - 08013 Barcelona

Concurso

Cálculo de intereses



VIC 2Ø + EXPANSION (8K o 16K)



Este programa nos lo envía **Jordi Pallarés Mulet**, desde Barcelona, advirtiéndonos que ha sido empleado en la enseñanza de alumnos de básica y BUP (aunque no nos dice si con éxito o no). El programa permite realizar cálculos con las fórmulas de interés simple y compuesto, definiendo el tiempo en años, meses o días y permitiendo la elección de cualquiera de las variables de la fórmula como incógnita. Todas estas opciones se pueden ir seleccionando de forma sencilla una vez que el programa está en ejecución sin más que responder a las preguntas que aparecen en la pantalla. Además, antes de proceder a realizar ningún cálculo, el programa presenta en pantalla las fórmulas que va a utilizar explicando brevemente los conceptos incluidos en las mismas.

El programa está escrito en BASIC estandar, sin ninguna referencia a posiciones específicas de memoria, por lo que puede ser empleado tanto para el **VIC** como para el **C-64**, teniendo en cuenta, eso sí, que está pensado para el formato de pantalla del **VIC** y que de utilizarlo con este ordenador es necesaria una ampliación de memoria (de 8K o 16K).

```
CAPITAL INICIAL=
 2000000   PTAS
CAPITAL FINAL=
 3498012.51   PTAS
REDITO= 15   %
TIEMPO= 4   ANOS
INTERES= 1498012.51  P
TAS
READY.
```

El programa viene estructurado en una serie de subrutinas de las que reseñamos las más importantes:

SUBRUTINAS

95ØØ — 952Ø	Rutina de impresión para la selección de parámetro desconocido
12ØØØ — 12Ø1Ø	Si el parámetro es desconocido vuelve a preguntar
1ØØØØ — 1ØØ2Ø	Entrada de datos
7ØØØ — 7Ø3Ø	Entrada de todos los datos menos el interés (incógnita)
7Ø5Ø — 7Ø8Ø	Entrada de todos los datos menos el capital
71ØØ — 713Ø	Entrada de todos los datos menos el rédito
715Ø — 718Ø	Entrada de todos los datos menos el tiempo
8ØØØ — 8Ø6Ø	Escritura y cálculo del capital final
8ØØ3 — 8ØØ5	Redondeo
6ØØØ — 6Ø7Ø	Escritura y cálculo del interés compuesto

Concurso

```
1 REM****************
2 REM
3 REM PROGRAMA DE CALCULO DE INTERESES
4 REM
5 REM***************
6 REM
10 REM JORDI PALLARES
11 REM
12 REM****************
13 N=1
15 PRINT"J"
20 PRINT"CON QUE TIPO DE       INTERES DESEA TRABAJARSIMPLE O COMPUESTO?"
30 INPUTA$
35 IF A$<>"SIMPLE"ANDA$<>"COMPUESTO"GOTO15
40 IF A$="COMPUESTO"THEN2000
50 PRINT"J"
52 REM***************
55 PRINT"CON QUE PERIODO DE     TIEMPO DESEA TRABAJAR:"
56 PRINT"-ANOS":PRINT
57 PRINT"-MESES":PRINT
58 PRINT"-DIAS":PRINT:PRINT:PRINT
59 PRINT"RECUERDE QUE COMERCIALMENTE:"
60 PRINT:PRINT:PRINT"1 MES=30 DIAS":PRINT:PRINT"1 ANO=360 DIAS"
61 INPUTB$
70 IF B$="MESES"THEN600
80 IF B$="DIAS"THEN1000
85 IF B$<>"ANOS"ANDB$<>"MESES"ANDB$<>"DIAS"GOTO50
90 REM CALCULO TRABAJANDO CON ANOS
95 PRINT"J"
100 PRINT"RECUERDE QUE TRABAJO  CON LA FORMULA:":PRINT:PRINT:PRINT:PRINT:PRINT
105 PRINT"I=(C*R*T(ANOS))/100":PRINT:PRINT:PRINT: PRINT:PRINT"DONDE:":PRINT
106 PRINT"C=CAPITAL","R=REDITO"
107 PRINT:PRINT"T=TIEMPO(EN ANOS)"
108 FORA=0TO 10000:NEXT
110 GOSUB9500
120 INPUTC$
130 IF C$="CAPITAL"THEN300
140 IF C$="REDITO"THEN400
150 IF C$="TIEMPO"THEN500
151 GOSUB12000
152 REM CALCULO CUANDO DESCONOCEMOS EL INTERES
160 GOSUB10000
170 GOSUB7000
200 REM CALCULO DEL INTERES
210 I=(C*R*T)/100
220 GOSUB 8000
230 END
300 REM CALCULO CUANDO DESCONOCEMOS EL CAPITAL
310 GOSUB10000
320 GOSUB7050
350 REM CALCULO DEL CAPITAL
360 C=(100*I)/(R*T)
370 GOSUB 8000
380 END
400 REM CALCULO CUANDO DESCONOCEMOS EL REDITO
410 GOSUB 10000
420 GOSUB7100
```

```
450 REMCALCULO DEL REDITO
460 R=(100*I)/(C*T)
470 GOSUB8000
480 END
500 REM CALCULO CUANDO DESCONOCEMOS EL TIEMPO EN ANOS
510 GOSUB10000
520 GOSUB7150
530 REM CALCULO DEL TIEMPO EN ANOS
540 T=(100*I)/(C*R)
550 GOSUB8000
560 END
600 REM CALCULO TRABAJANDO CON MESES
609 PRINT"□"
610 PRINT"RECUERDE QUE TRABAJO  CON LA FORMULA:":PRINT:PRINT:PRINT
616 PRINT:PRINT:PRINT"  I=(C*R*T)/1200":PRINT:PRINT
617 PRINT: PRINT:PRINT"DONDE:":PRINT
618 PRINT"C=CAPITAL","R=REDITO":PRINT:PRINT "T=TIEMPO(EN MESES)"
619 FOR A=0TO10000:NEXT
620 GOSUB9500
630 INPUTC$
640 IF C$="CAPITAL"THEN810
650 IF C$="REDITO"THEN910
660 IF C$="TIEMPO"THEN960
670 GOSUB12000
700 REM CALCULOS PARA EL TIEMPO EN MESES
710 REM CALCULO CUANDO DESCONOCEMOS EL INTERES
720 GOSUB10000
730 GOSUB7000
740 I=(C*R*T)/1200
750 GOSUB8000
760 END
800 REM CALCULO CUANDO DESCONOCEMOS EL CAPITAL
810 GOSUB10000
820 GOSUB7050
830 C=(1200*I)/(R*T)
840 GOSUB8000
850 END
900 REM CALCULO CUANDO DESCONOCEMOS EL REDITO
910 GOSUB10000
920 GOSUB7100
930 R=(1200*I)/(C*T)
940 GOSUB8000
950 END
960 REM CALCULO CUANDO DESCONOCEMOS EL TIEMPO
965 GOSUB10000
970 GOSUB7150
975 T=(1200*I)/(C*R)
980 GOSUB8000
985 END
1000 REM CALCULO TRABAJANDO CON DIAS
1010 PRINT"□"
1011 PRINT"RECUERDE QUE TRABAJO  CON LA FORMULA:":PRINT:PRINT:PRINT:PRINT
1012 PRINT"  I=(C*R*T)/36000":PRINT:PRINT:PRINT:PRINT"DONDE:":PRINT
1013 PRINT"C=CAPITAL","R=REDITO":PRINT: PRINT"T=TIEMPO(EN DIAS)"
1020 FORA=0TO 10000:NEXT
1030 GOSUB9500
1040 INPUTC$
```

PROGRAMA COMPROBADO commodore Magazine



¡¡Exclusivo para lectores de 

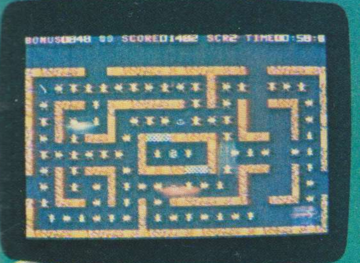


CLUB DE SOFTWARE



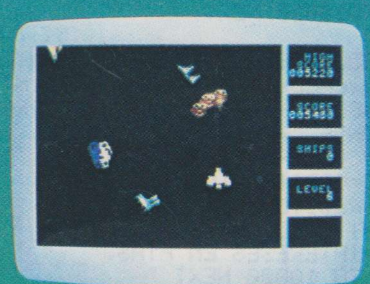
Ha seleccionado para usted cuatro excelentes programas de juego para su ordenador **COMMODORE 64.**

Diviértase con ellos y, **además, ahorre dinero** con esta oferta única y exclusiva para los lectores de esta revista.



HIDEOUS BILL

¿Está Bill (es decir, tú) dispuesto a luchar contra los gigantes para salvar a Greta, su (tú) único y verdadero amor? Si no lo logras, será comida por las hormigas. Y no olvides el peligro del barro venenoso, a no ser que lleves contigo una tableta de antídoto. Puedes valerte de los arpones, pero durarán poco. Rompiendo paredes abrirás caminos para la fuga. Al final, el amor y una sorpresa.
Precio: 1.550 ptas.



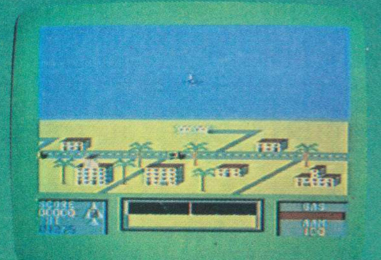
AMBUSH

Estás pilotando una nave espacial y tienen que cargar combustible de una nave nodriza. En el camino serás atacado. Puedes utilizar tu cañón de rayo laser, pero cuando un atacante es alcanzado no muere: se divide en dos y tu problema se multiplica. No acaban allí tus dificultades: tienes que esquivar (o destruir) sus bombas de protones. Hay siete, ¡siete! oleadas de enemigos. Tal vez te convenga atracar en la nave nodriza para coger un cañón laser más potente.
Precio: 1.550 ptas.



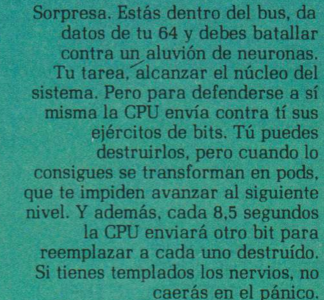
FALCON PATROL

Como único superviviente de la patrulla Halcón, te toca a tí conseguir que los reactores atacantes no destruyan las instalaciones petroleras y tus bases de aprovisionamiento. Estás armado con cien misiles y un radar muy sofisticado en pantalla. Si te quedas sin combustible, podrás aterrizar verticalmente, pero mientras estés en tierra serás vulnerable a la metralla. Cuanto más tiempo sobrevivas, más enemigos vendrán a por tí.
Precio: 1.550 ptas.



BITMANIA

Sorpresa. Estás dentro del bus, da datos de tu 64 y debes batallar contra un aluvión de neuronas. Tu tarea, alcanzar el núcleo del sistema. Pero para defenderse a sí misma la CPU envía contra tí sus ejércitos de bits. Tú puedes destruirlos, pero cuando lo consigues se transforman en pods, que te impiden avanzar al siguiente nivel. Y además, cada 8,5 segundos la CPU enviará otro bit para reemplazar a cada uno destruído. Si tienes templados los nervios, no caerás en el pánico.
Precio: 1.550 ptas.





Oferta especial 4X3

Usted puede comprar cualquiera de estos cassettes a su precio individual de 1.550 Ptas. Pero también puede adquirir el lote de 4 por sólo 4.650 Ptas. Y AHORRARSE 1.550 Ptas. O sea, cuatro cassettes por el precio de tres. Aproveche esta oferta limitada.

Recorte y envíe este cupón HOY MISMO a:
INFODIS, S.A.
Bravo Murillo, 377-5.º-A
28020 MADRID

CUPON DE PEDIDO

Envíenme a mi domicilio, al precio de 1.550 Ptas. cada una, más 95 Ptas. de gastos, la cassette o cassettes que le indico a continuación, marcando con una (X) en el casillero correspondiente. Queda bien entendido que pidiendo las 4 cassettes obtengo un ahorro de 1.550 Ptas., lo que me supone UNA CASSETTE GRATIS **(Sólo pagaré 4.650 Ptas. más 95 Ptas. por gastos de envío).**

☐ **HIDEOUS BILL** ☐ **AMBUSH** ☐ **FALCON PATROL** ☐ **BITMANIA**

☐ Las 4 cassettes, en las ventajosas condiciones de AHORRO para mí **(4.650 Ptas.).**

El importe lo abonaré:
POR CHEQUE ☐ **CONTRA REEMBOLSO** ☐ **CON TARJETA DE CREDITO** ☐
American Express ☐ **Visa** ☐ **Interbank** ☐ Fecha de caducidad ____________

Número de mi Tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Nombre ____________

Dirección ____________

Ciudad ____________ D.P. ______

Provincia ____________

Cada envío llevará 95 Ptas. en concepto de gastos.

Concurso

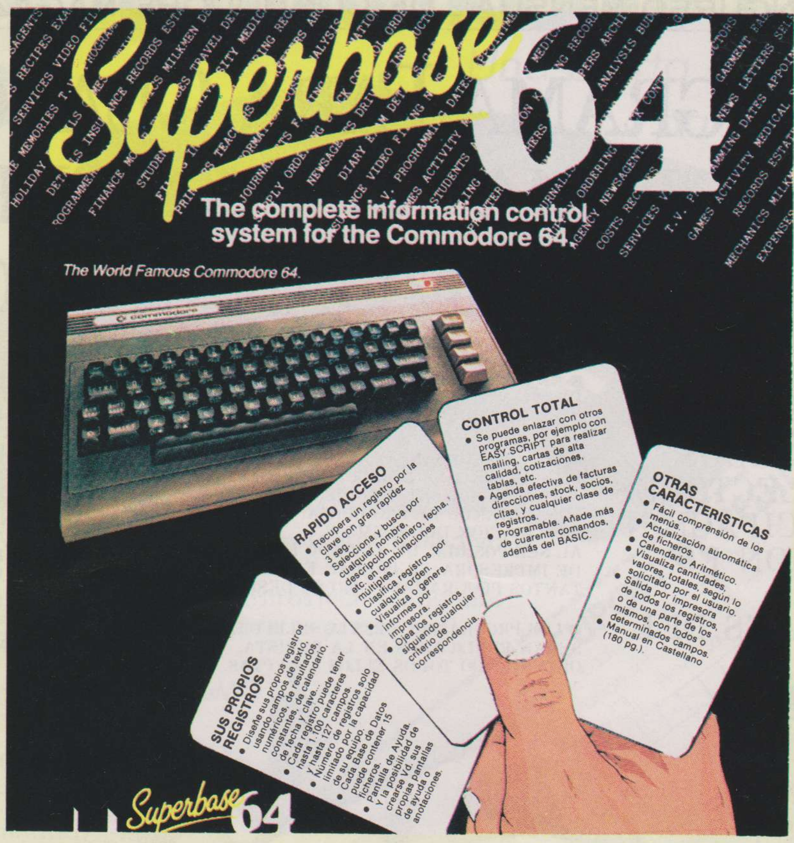
```
1050 IF C$="CAPITAL"THEN 3110
1060 IF C$="REDITO"THEN3210
1070 IF C$="INTERES"THEN3000
1080 IF C$="TIEMPO"THEN3400
1090 GOSUB12000
2000 REM CALCULOS PARA INTERESES COMPUESTOS
2010 PRINT"J"
2015 PRINT:PRINT:PRINT:PRINT:PRINT"TRABAJO CON ANOS"
2016 PRINT"RECUERDE QUE UN ANO   COMERCIAL=360 DIAS":PRINT:PRINT
2018 FOR I=0TO10000:NEXT
2019 PRINT"J"
2020 PRINT"RECUERDE QUE TRABAJO  CON LAS FORMULAS:":PRINT:PRINT
2021 PRINT"    C=CO(1+R)↑N":PRINT:PRINT
2022 PRINT"I=CO[(1+(R/100))↑N -1]":PRINT
2030 PRINT"DONDE:":PRINT
2031 PRINT"C=CAPITAL FINAL":PRINT
2032 PRINT"CO=CAPITAL INICIAL":PRINT
2033 PRINT"R=REDITO EN %":PRINT
2034 PRINT"N=TIEMPO EN ANOS":PRINT
2035 PRINT"I=INTERESES EN PTAS."
2037 FOR A=0 TO 10000:NEXT
2040 PRINT"J"
2050 PRINT"CUAL ES EL VALOR DESCONOCIDO,":PRINT:PRINT:PRINT
2051 PRINT"-CAPITAL FINAL":PRINT
2052 PRINT"-CAPITAL INICIAL":PRINT
2053 PRINT"-REDITO":PRINT
2054 PRINT"-TIEMPO":PRINT
2055 PRINT"-INTERES?":PRINT
2060 INPUTD$
2070 IFD$="CAPITAL FINAL"THEN2120
2080 IFD$="CAPITAL INICIAL"THEN2210
2090 IF D$="REDITO" THEN  2300
2100 IF D$="TIEMPO"THEN2400
2110 IF D$="INTERES"THEN2500
2120 REM CALCULO DEL CAPITAL FINAL
2130 GOSUB10000
2140 INPUT"CAPITAL INICIAL=";CO
2150 INPUT"REDITO=";Z
2155 R=Z/100.
2160 INPUT"TIEMPO(EN ANOS)=";N:PRINT
2170 C=CO*(1+R)↑N
2180 GOSUB 6000
2190 END
2200 REM CALCULO DEL CAPITAL INICIAL
2210 GOSUB10000
2220 INPUT"CAPITAL FINAL";C
2230 INPUT"REDITO";Z
2240 INPUT"TIEMPO(EN ANOS)";N
2245 R=Z/100.
2250 CO=C/((1+R)↑N)
2260 GOSUB6000
2270 END
2300 REM CALCULO DEL REDITO
2310 GOSUB10000
2320 INPUT "CAPITAL INICIAL";CO
2330 INPUT"CAPITAL FINAL";C
```

```
2340 INPUT"TIEMPO(EN ANOS)";N
2350 R=((C/CO)↑(1/N))-1
2355 Z=R*100
2360 GOSUB6000
2370 END
2400 REM CALCULO DEL TIEMPO EN ANOS
2410 GOSUB 10000
2420 INPUT"CAPITAL INICIAL";CO
2430 INPUT"CAPITAL FINAL";C
2440 INPUT"REDITO";Z
2445 R=Z/100.
2450 N=(LOG(C/CO)/LOG(10))/(LOG(1+R)/LOG(10))
2460 GOSUB 6000
2470 END
2500 REM CALCULO DEL INTERES
2510 GOSUB10000
2520 INPUT"CAPITAL INICIAL";CO
2530 INPUT"REDITO";Z
2535 R=Z/100.
2540 INPUT "TIEMPO(EN ANOS)";N
2550 I=CO*(((1+R)↑(N))-1)
2555 C=CO*(1+R)↑N
2560 PRINT"□"
2564 REM REDONDEO A DOS DECIMALES
2565 CO=INT(CO*100+0.5)/100:C=INT(C*100+0.5)/100
2566 I=INT(I*100+0.5)/100
2570 PRINT"CAPITAL INICIAL=":PRINT:PRINTCO;" PTAS ":PRINT:PRINT
2575 PRINT"CAPITAL FINAL=":PRINT:PRINT C" PTAS ":PRINT:PRINT
2580 PRINT"REDITO=";Z;" % ":PRINT:PRINT
2590 PRINT"TIEMPO=";N;" ANOS' ":PRINT:PRINT
2600 PRINT"INTERES=";I;" PTAS "
2700 END
3000 REM CALCULO CUANDO DESCONOCEMOS EL INTERES
3010 GOSUB10000
3020 GOSUB7000
3030 I=(C*R*T)/36000
3040 GOSUB8000
3050 END
3100 REM CALCULO CUANDO DESCONOCEMOS EL CAPITAL
3110 GOSUB10000
3120 GOSUB7050
3130 C=(3600*I)/(R*T)
3140 GOSUB8000
3150 END
3200 REM CALCULO CUANDO DESCONOCEMOS EL REDITO
3210 GOSUB10000
3220 GOSUB7100
3230 R=(36000*I)/(C*T)
3240 GOSUB8000
3250 END
3400 REM CALCULO CUANDO DESCONOCEMOS EL TIEMPO
3410 GOSUB10000
3420 GOSUB7150
3430 T=(36000*I)/(R*C)
3440 GOSUB8000
3450 END
6000 REMSUB.ESCRITURA INTERES COMPUESTO Y CALCULO DEL INTERES
```

Concurso

PROGRAMA commodore COMPROBADO

```
6005 PRINT"♥"
6010 I=C-CO
6014 REM REDONDEO A DOS DECIMALES
6015 CO=INT(CO*100+0.5)/100:C=INT(C*100+0.5)/100:
6016 I=INT(I*100+0.5)/100
6020 PRINT"CAPITAL INICIAL=":PRINT
6025 PRINTCO;"PTAS.":PRINT
6030 PRINT"CAPITAL FINAL=":PRINT
6035 PRINT C;"PTAS.":PRINT
6040 PRINT"TIEMPO=";N;"ANOS":PRINT:PRINT
6050 PRINT"REDITO=";Z;"%":PRINT:PRINT
6060 PRINT"INTERES=";I;"PTAS."
6070 RETURN
7000 INPUT"CAPITAL=";C:PRINT
7010 INPUT"REDITO=";R:PRINT
7020 INPUT"TIEMPO=";T
7030 RETURN
7050 INPUT"INTERES=";I:PRINT
7060 INPUT"REDITO=";R:PRINT
7070 INPUT"TIEMPO=";T
7080 RETURN
7100 INPUT"INTERES=";I:PRINT
7110 INPUT"CAPITAL=";C:PRINT
7120 INPUT"TIEMPO=";T:PRINT
7130 RETURN
7150 INPUT"CAPITAL=";C:PRINT
7160 INPUT"INTERES=";I:PRINT
7170 INPUT"REDITO=";R
7180 RETURN
8000 REM SUB.DE ESCRITURA Y CALCULO DE CAPITAL FINAL
8002 PRINT"♥"
8003 REM REDONDEO A DOS DECIMALES DE LAS PTAS
8004 I=INT(I*100+0.5)/100:C=INT(C*100+0.5)/100:CF=INT(CF*100+0.5)/100
8005 CF=C+I
8010 PRINT"INTERES=";I;"PTAS.":PRINT:PRINT
8020 PRINT"CAPITAL=";C;"PTAS.":PRINT:PRINT
8030 PRINT"REDITO=";R;"%":PRINT:PRINT
8040 PRINT"TIEMPO=";T;B$:PRINT:PRINT
8050 PRINT"CAPITAL FINAL=":PRINT:PRINT CF;"PTAS.":PRINT:PRINT
8060 RETURN
9500 PRINT"♥":PRINT:PRINT
9510 PRINT"CUAL ES EL VALOR       DESCONOCIDO,":PRINT
9511 PRINT"-INTERES":PRINT
9512 PRINT"-CAPITAL":PRINT
9513 PRINT"-REDITO":PRINT
9514 PRINT"-TIEMPO?"
9520 RETURN
10000 PRINT"♥"
10005 PRINT:PRINT:PRINT:PRINT
10010 PRINT"ENTRAR LOS VALORES     CONOCIDOS CONFORME SE LOS VAYA PIDIENDO"
10015 PRINT
10020 RETURN
12000 IFC$<>"INTERES"ANDC$<>"CAPITAL"ANDC$<>"REDITO"ANDC$<>"TIEMPO"GOTO110
12010 RETURN
```



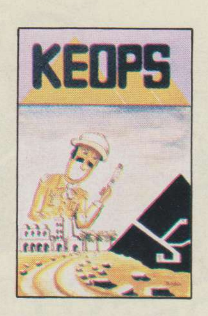
P.V.P. 22.500 Ptas. (disco)



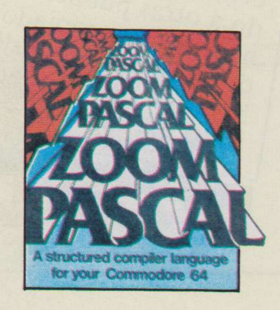
Elimina variantes y figuras. Desarrollo, lista por impresora los boletos del patronato o en papel continuo para archivo, introduciendo el boleto premiado hace el escrutinio de la jornada imprimiendo los premios dándole el número de boleto y columna, etc.
P.V.P. 15.000 Pts.



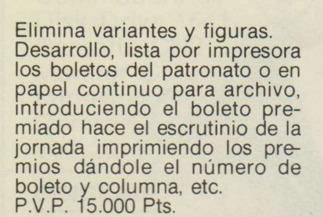
Usa el mismo intérprete que el Basic del ordenador. Mantiene la misma memoria libre y velocidad de ejecución.
Comandos y mensajes en castellano.
P.V.P. 2.000 Ptas. cinta
P.V.P. 2.500 Pts. disco



Maravilloso juego de aventuras. Deberá emplear su inteligencia y su ingenio.
P.V.P. 1.500 Ptas. cinta
P.V.P. 2.000 Ptas. disco



Zoom-Pascal para CBM-64 extremadamente rápido. Añade extensiones y funciones que lo hacen verdaderamente útil.
P.V.P. 12.000 Pts.
P.V.P. 12.500 Pts.



CONTROL DE ALMACEN

Controla 1.200 referencias de almacén, en unidades, kilos, litros, metros, etc. Stock mínimo. Realiza gestiones de entrada y salida (incluidas devoluciones). Podrá hacer altas, bajas, modificaciones y consultas, Proceso de Inventario Físico, Inventario a Precio de Costo y a Precio de Venta. Etiquetas...
Le dará en cualquier momento un listado de productos sin movimiento. Además tiene una opción de anotación, para los gastos de su empresa, con análisis de rentabilidad de su negocio, sin pasar por contabilidad. De fácil manejo. TODO EN CASTELLANO.
P.V.P. en disco 35.000 Pts.

SAKATI, S.A. Empresa Líder en Soft para COMMODORE-64, tiene un amplio catálogo de programas, donde Vd. encontrará la solución a sus necesidades.

¡CONSULTENOS!

BOLETIN DE PEDIDO

A enviar a SAKATI, S.A. - Ardemans, 24 - 28028-MADRID

Ref.	Cdad.	Precio

☐ Talón adjunto ☐ Contra reembolso TOTAL ______

Fecha: Firma:

Don ______

Calle ______ N.º ______

Código Postal ______ Ciudad ______

Provincia ______ Tel. ______

SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA. LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A: CONCURSO COMMODORE MAGAZINE

ORIGINALIDAD

Los programas han de ser inéditos. No deben haberse enviado a otras publicaciones, ni ser copiados de manuales y/o libros, ya sean españoles o extranjeros.



commodore Magazine

C/Bravo Murillo, 377. 28002-Madrid

Protege tu Commodore con esta Práctica Funda

Una oferta especial y exclusiva para nuestros lectores

SOLO **875** PTAS.

Una práctica funda lavable y resistente, que protegerá del polvo y de otros deterioros a tu COMMODORE.

¡Y que por ser una oferta exclusiva para nuestros lectores puedes conseguirla con un 25% de descuento sobre su precio real de venta!

¡Apresúrate! Recorta y envía HOY MISMO este cupón a:

COMMODORE MAGAZINE (FUNDAS)
Bravo Murillo, 377 - 28020-MADRID

Las existencias son limitadas
¡No te quedes sin ella!

CUPON DE PEDIDO

Sí, envíenme al precio de 875 Ptas. cada una, más 100 Ptas. de gastos de envío, ______ fundas para mi COMMODORE. El importe lo abonaré: Contra reembolso ☐ Con mi tarjeta de crédito ☐

Adjunto cheque ☐ American Express ☐ Visa ☐ Interbank ☐

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Fecha de caducidad ______

NOMBRE ______

DIRECCION ______

CIUDAD ______

PROVINCIA ______

Gastos de envío por cada pedido: 100 Ptas.

ELECTROAFICIÓN COMPUTER

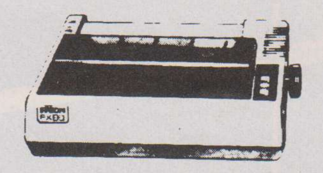
C/VILLARROEL, 104 BARCELONA-11 TLF. 2537600-09

PRODUCTOS COMMODORE



- Commodore-64
- Disk Drive 1541
- Cassette CN2
- Monitor Color 1701
- Impresora MPS-801
- Commodore 64SX Portable
- VIC-20

IMPRESORAS



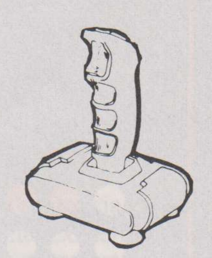
- Seikosha
- Star
- Epson
- NewPrint
- C. Itoh
- Riteman

SINCLAIR



- Spectrum 48K
- Impresora Seikosha con interface
- Microdrive
- Teclado DK'TRONICS
- LAPIZ óptico
- Amplificador Sonido

GAMA COMPLETA DE ACCESORIOS

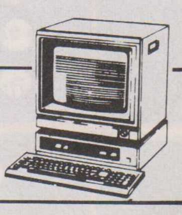


- Interfaces
- Joysticks
- Sintetizadores de voz
- Cassettes
- Cintas
- Discos
- Base de Datos
- Easy Script
- Monitores
- Interpod
- Cables
- Procesador de Textos
- Libros

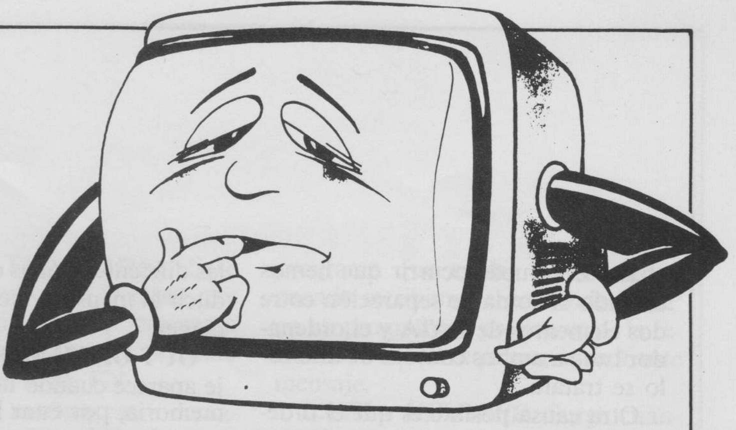
SOFTWARE

- Contabilidad
- Contabilidad Doméstica
- Control de Stocks
- Mailing y Etiquetas
- Ficheros
- Base de Datos
- Gran variedad de Juegos
- Programas Educativos

ORDENADORES DE GESTION



- Pal Computer
- Commodore
- Apple



Identifica tus errores

Durante el desarrollo de un programa es corriente que cometamos algún error de programación. Esto también puede ocurrir mientras transcribimos un programa, por ejemplo, de **Commodore Magazine**. En el momento en que el ordenador detecta algo fuera de orden, se lo advierte rápidamente al usuario por medio de un mensaje de error en la pantalla. Esto representa una ventaja con respecto a diversos sistemas de diseño más antiguo, que solamente proporcionan un número correspondiente al mensaje de error y, posteriormente debemos contrastarlo con el Manual.

De cualquier forma, es fundamental conocer el significado del mensaje, para ir directamente a atajar la causa que lo produjo. Este artículo se detiene en la descripción de los mensajes de manera más o menos detallada.

Comencemos por los errores que afectan a los ficheros de almacenamiento de programas y datos en *cassette* o *diskette*.

FILE OPEN. Hace acto de aparición cuando pretendemos abrir un nuevo fichero, con ayuda del comando OPEN, pero resulta que ese fichero ya ha sido abierto. Entonces dos cosas pueden ocurrir: bien no necesitamos reabrir el fichero o necesitamos un número distinto de fichero lógico con el comando. En caso de duda, recurra a cerrar (CLOSE) el fichero y volver a abrirlo (OPEN).

FILE NOT FOUND. Cuando estamos tratando de cargar (LOAD) o verificar (VERIFY) un fichero almacenado en cinta o *diskette* y el ordenador no le localiza. Lo más probable es que el fichero no se encuentre allí o hayamos escrito mal el nombre del mismo.

FILE NOT OPEN. Cuando utilizamos alguno de los comandos que implican la apertura previa de un fichero, aparece este mensaje si no lo ha sido. Los comandos que lo pueden provocar son: CMD, CLOSE, GET#, INPUT#, PRINT#.La solución consite en utilizar nuevamente OPEN y volver a intentar el comando.

DEVICE NOT PRESENT. Normalmente, este mensaje suele denunciar que un dispositivo al que nos referimos no ha sido conectado al ordenador, o si lo está no recibe alimentación eléctrica (recuerde comprobar el conmutador de puesta en marcha). En realidad viene a decir que un dispositivo de Entrada/Salida no está disponible para ser utilizado con los comandos OPEN, CLOSE, CMD, PRINT#, INPUT# o GET#. Algunas veces ocurre que el mensaje no responde a una causa aparente, sobre todo cuando se trabaja con la unidad de *diskette*. Aunque es poco corriente que ocurra esto, la mejor solución siempre ha parecido ser la puesta a cero del sistema, y la manera más rápida es apagar y volver a conectarlo.

NOT INPUT FILE. Si hemos abierto un fichero al que sólo queremos enviar datos y tratamos de leer algo desde él, surge el mensaje, es decir, cuando empleamos GET# o INPUT#. Compruebe el tercer parámetro que va detrás del OPEN correspondiente y si no es cero o no hay, el fichero abierto es únicamente de escritura.

NOT OUPUT FILE. Se trata del caso complementario al anterior. Si hemos abierto un fichero para entrada de datos e intentamos escribir en él, obtenemos el mensaje. En este supuesto también podemos examinar el parámetro situado en tercer lugar, tras OPEN, y si no es 1 ó 2, el fichero es sólo para escribir en él.

NEXT WITHOUT FOR. Aparece cuando el ordenador encuentra una sentencia NEXT y no ha localizado previamente un FOR, que dé comienzo al bucle. Esto suele ocurrir más a menudo cuando encadenamos varios bucles, unos dentro de los otros, y no tenemos cuidado de emparejarlos.

SINTAX. Salta siempre que el ordenador se encuentra con un término o construcción ilegal, es decir, la sintaxis de escritura del texto del programa BASIC no es correcta. Normalmente la culpabilidad hay que buscarla en la forma de teclear las palabras clave que utiliza el BASIC, pues el intérprete es sumamente crítico a la hora de leerlas. Podríamos habernos comido una letra o añadido alguna de más. También es conveniente fijarnos en el número de paréntesis abiertos y cerrados, que debe ser el mismo.

RETURN WITHOUT GOSUB. Ocurre cuando el ordenador localiza un RETURN que no va asociado a un GOSUB previo.

En muchos casos podría ocurrir que el programa hubiera entrado en una subrutina sin que hubiera intervenido GOSUB. En tal caso lo recomendable es examinar los pasos de programa que tienen posibilidad de alcanzar a la subrutina. Es muy socorrido utilizar END al final del programa, antes de entrar en la zona destinada a las subrutinas, que se suele situar al final del listado.

La existencia de GOTOs incontrolados también puede derivar a una situación de este tipo.

OUT OF DATA. Este mensaje está relacionado con DATA y READ. Cada vez que el programa llega a una sentencia READ, el ordenador va al siguiente elemento de una sentencia DATA, señalado por un puntero interno del ordenador. Si hay más datos que leer en sentencias READ de los que realmente existen en las DATA, es cuando surge este problema. Por tanto es conveniente contar en el listado para comprobar que disponemos de los elementos suficientes. Mu-

chas veces puede ocurrir que hemos omitido la coma de separación entre dos elementos de DATA y el ordenador trata a ambos como si de uno solo se tratara.

Otra causa posible es que el ordenador necesite volver a leer todos los elementos DATA y no hayamos hecho utilización de RESTORE para inicializar el puntero interno.

ILLEGAL QUANTITY. Si utilizamos una constante o variable situada fuera del rango que puede admitir el ordenador, surge este mensaje. Por ejemplo, si en PEEK utilizamos un valor que supere los K de memoria límite del ordenador o queremos introducir con POKE un número mayor de 255 en una dirección de memoria.

OVERFLOW. Si realizamos un cálculo con el ordenador que supere la cantidad 1.70141884E+38 o sea menor que 1.70141884E-38, obtenemos este mensaje de desbordamiento. En este tipo de programas es útil disponer una sentencia que controle las diferentes etapas que podrían conducir al mensaje, mediante comparaciones.

OUT OF MEMORY. Este mensaje aparece cuando nos quedamos sin memoria, por estar llena toda la disponible. Se puede deber principalmente a que el programa es demasiado largo o a que está creando demasiadas nuevas variables. Aquí la solución pasa obligatoriamente por añadir más memoria, en el caso del **Vic**, o depurar el programa para que haga las mismas cosas con menores requerimientos de almacenamiento.

Es posible que una vez que consultamos cuanta memoria nos queda, mediante el comando FRE, y la respuesta es favorable, nos aparezca el mensaje. En tal caso suele suceder que estamos empleando demasiados bucles FOR...NEXT o GOSUBs encadenados entre sí. Otra ocasión en la que es frecuente encontrar este mensaje es cuando cargamos un programa que está almacenado en *cassette*. El programa va precedido por un cabecero, que incorpora determinada información vital para el correcto almacenamiento del programa que le sigue, así como el nombre del fichero que lo guarda. Si esta información resulta alterada o no se carga correctamente en el ordenador, puede alterar parte del sistema operativo del ordenador. La solución suele pasar casi siempre por la conexión y vuelta a conectar del ordenador.

UNDEF'D STATEMENT. Con esta leyenda el ordenador nos advierte que no existe tal número de línea. Suele ocurrir después de que tecleemos distintos comandos para la ejecución inmediata de un programa, tal como RUN un número de línea, GOTO o GOSUB.

BAD SUBSCRIPT. Cuando queremos definir una matriz que almacenará datos numéricos o cadenas, lo primero que hemos de hacer es dimensionar esa matriz. Si posteriormente, en el programa, nos referimos a un elemento que sale fuera de esa definición es cuando surge el mensaje. En otras palabras, el error se produce cuando el subíndice de ese elemento no esta comprendido en el conjunto de los que puede albergar la matriz.

REDIM'D ARRAY. Cuando definimos una matriz con DIM, debemos darle un nombre que la caracterice, pero ese nombre solamente podemos utilizarlo una vez. Si lo volvemos a utilizar surgirá este mensaje. También podría tener lugar si introdujéramos un DIM dentro de un bucle, pues la matriz sería definida tantas veces como se repita el mismo.

DIVISION BY ZERO. Como es lógico, el ordenador no nos permite que dividamos ningún valor por cero. Podría ocurrir en el transcurso de una operación que utiliza valores proporcionados por otras partes del programa. En tal caso hay que repasarlo para ver dónde se genera ese valor cero.

ILLEGAL DIRECT. Muchas de las **palabras claves** del BASIC no pueden ser utilizadas como comandos ejecutables directamente. Si lo inten-

DIRECCION MEMORIA	NUMERO DE MENSAJE	MENSAJE
41373	1	TOO MANY FILES
41387	2	FILE OPEN
41396	3	FILE NOT OPEN
41409	4	FILE NOT FOUND
41423	5	DEVICE NOT PRESENT
41441	6	NOT INPUT FILE
41455	7	NOT OUTPUT FILE
41470	8	MISSING FILE NAME
41487	9	ILLEGAL DEVICE NUMBER
41508	10	NEXT WITHOUT FOR
41524	11	SYNTAX
41530	12	RETURN WITHOUT GOSUB
41550	13	OUT OF DATA
41561	14	ILLEGAL QUANTITY
41577	15	OVERFLOW
41585	16	OUT OF MEMORY
41598	17	UNDEF'D STATEMENT
41615	18	BAD SUBSCRIPT
41628	19	REDIM'D ARRAY
41641	20	DIVISION BY ZERO
41657	21	ILLEGAL DIRECT
41671	22	TYPE MISMATCH
41684	23	STRING TOO LONG
41699	24	FILE DATA
41708	25	FORMULA TOO COMPLEX
41727	26	CAN'T CONTINUE
41741	27	UNDEF'D FUNCTION
41757	28	VERIFY
41763	29	LOAD

tásemos salta este mensaje a la pantalla. En el caso de que fuera imprescindible emplear uno, deberíamos desarrollar una línea o un programa breve, que nos permita la aplicación casi inmediata del comando deseado al ordenador.

TYPE MISMATCH. Cuando el programa intenta poner el tipo incorrecto de dato en una varible, obtenemos este mensaje. Por ejemplo, podríamos encontrar que se asignará una cadena a una variable numérica, entera o en punto flotante.

Una revisión del programa dará al traste con el problema.

STRING TOO LONG. La traducción literal es suficientemente gráfica: Cadena demasiado larga. La máxima longitud permitida a una cadena es de 255 caracteres (bytes). Este mensaje puede aparecer cuando intentamos concatenar (unir) cadenas y el resultado es una que supera esta longitud máxima.

Cuando un programa ha de llevar a cabo operaciones con cadenas sería conveniente utilizar controles que eviten este tipo de error, obviamente con el empleo de la **palabra clave** LEN$, que nos va proporcionando los valores numéricos de la longitud de las cadenas.

FORMULA TOO COMPLEX. El ordenador tiene determinados límites a la hora de calcular expresiones dipuestas a manera de fórmula. Si ésta es demasiado compleja ante los criterios de la máquina, es cuando se produce el mensaje. Uno de los factores que más influyen suele ser la existencia de un número elevado de paréntesis, por tanto, una posible solución pasaría por descomponer la fórmula en pasos que conduzcan al mismo resultado.

CAN'T CONTINUE. Este mensaje aparece después de que utilicemos el comando CONT, para hacer que un programa prosiga ejecutando a partir del punto en que se detuvo. Este comando necesita para trabajar que el programa hubiera corrido antes de pararse.

UNDEF'D FUNCTION. Este mensaje es visualizado siempre que intentemos utilizar en un programa una función que no hemos definido previamente con DEF FN.

VERIFY. En muchos casos deseamos verificar que lo grabado en la cinta o *diskette* es exactamente igual que el programa existente en la memoria del ordenador. Si la identidad no es absoluta, este mensaje nos lo advierte. La mejor solución consiste en volver a grabar (SAVE) el programa de nuevo.

LOAD. Indica un error surgido durante el proceso de carga de un programa en la memoria del ordenador, a partir de un cinta o *diskette*. Lo aconsejable es volver a intentarlo de nuevo, después de haber apagado y vuelto a encender el ordenador. Si tras intentarlo varias veces persiste el problema, poco nos queda por hacer, excepto posicionar de manera diferente el *cassette* con respecto al televisor; pudiera ser que algún campo eléctrico alterase el proceso de la carga.

BAD DATA. El ordenador esperaba recibir datos numéricos y el programa le proporciona una cadena de caracteres o viceversa. Examinar las variables que siguen a las **palabras clave** READ e INPUT, para comprobar de qué tipo son.

BREAD IN LINE. Indica el número de línea en que se ha detenido un programa que estaba ejecutándose, cuando escribimos el comando STOP en un programa o presionamos la tecla RUN/STOP es proporcionado este mensaje.

EXTRA IGNORED. Surge cuando introducimos un número elevado de elementos como respuesta a una sentencia INPUT.

FILE ALREDY EXISTS. En un disco no podemos almacenar más de un fichero con el mismo nombre. La solución consiste en utilizar uno distinto.

```
1000 OPEN3,4
1005 PRINT#3,CHR$(15)"DIRECCION    NUMERO      MENSAJE"
1006 PRINT#3,CHR$(15)"MEMORIA      DE                 "
1007 PRINT#3,CHR$(15)"             MENSAJE"
1008 PRINT#3,CHR$(15),CHR$(10)
1010 A=41373
1020 X=0
1030 X=X+1
1040 PRINT#3,CHR$(15)A;"           ";X;"       ";
1050 A=A+1
1060 B=PEEK(A)
1070 IF B>128 THEN GOTO 2000
1080 C$=CHR$(B)
1090 PRINT#3,CHR$(15)C$;
1100 IF A=203 THEN CLOSE3
1110 GOTO 1050
2000 B=B-128
2010 C$=CHR$(B)
2020 PRINT#3,CHR$(15)C$
2030 GOTO 1030
```

REDO FROM START. Esto significa que hemos introducido el tipo erróneo de datos cuando el ordenador nos los reclama en una sentencia INPUT.

La explicación de estos mensajes puede ayudarnos a compreder mejor lo que trata de comunicarnos el ordenador. Su correcta interpretación es una poderosa herramienta para depurar los errores cometidos durante la programación.

El listado de la figura nos proporciona, a título de curiosidad, una forma de ver cómo están almacenados los mensajes en la memoria del ordenador, con las posiciones de memoria en que se buscan a algunos de ellos en el caso de que surga la ocasión. Este programa ha sido desarrollado para el **64**.

Trucos

DEMOSTRACION SOBRE COMO PODEMOS PARTIR LA PANTALLA Vic-20

Esta corta rutina divide la pantalla del **Vic-20** en dos secciones. Mediante los POKE de las líneas 30 y 40 podemos variar los colores del fondo y el reborde, así como la posición por donde partimos la pantalla.

Una vez que lo hemos cargado, el código máquina que contiene es totalmente transparente a los programas en BASIC que introduzcamos, pues en realidad este programa no es ni más ni menos que una forma sencilla de introducir programas en código máquina de manera más cómoda.

Una vez cargado, queda depositado en la memoria del **Vic-20**, pudiendo borrar el programa si deseamos.

En realidad esto es una demostración, pero puede ser un interesante punto de partida para desarrollar otras ideas.

```
10 FORJ=828TO895:READX:POKEJ,X:NEXT
20 SYS828
30 PRINT"JPOKE869,CONTROLA         LA POSICION DONDE PARTIMOS;
40 PRINT"POKE876 AND POKE889  CONTROLA FONDOS Y        REBORDES.
50 DATA 173,4,144,208,251,169,43,141
60 DATA 37,145,169,66,141,36,145,169
70 DATA 1,141,94,3,120,169,3,141,21,3,169,93
80 DATA 141,20,3,88,96,162,0,240,18,206,94,3,169,78,205,4,144
90 DATA 208,251,169,170,141,15,144,76,21,235,169,1,141,94,3,169
100 DATA153,141,15,144,76,191,234
```

Envianos la foto de tu ordenador



Esta es la foto que nos hace llegar **Jesús Rifá** desde Barcelona, para mostrarnos a todos cómo el ingenio supera las difíciles barreras impuestas por la falta de espacio.

Podéis ver que una instalación compacta puede albergar a todos los elementos necesarios para un mayor disfrute del ordenador. Seguidamente, él mismo nos comenta los pormenores: «*Como pueden observar, mi espacio para el ordenador es bastante reducido, lo tengo dentro del armario con un estante corredizo, que lo puedo meter y sacar para poder trabajar mejor. La instalación eléctrica pasa por detrás del armario, donde hay un interruptor, un punto de luz y una base de enchufe con un triple, donde lo conecto todo. Cuando cierro el interruptor se apaga todo. Tengo conectado el cassette* Commodore, *un joystick y el monitor es una televisión de blanco y negro.*»

Pues enhorabuena por tu magnífica idea. Pronto recibirás en premio un *cassette* de juegos para tu **64**.



No se caliente la "CABEZA"

SEIKOSHA

IMPRESORAS

Nuestra calidad es "SEIKO"; nuestros precios, únicos. Si desea más información, consulte con nuestro distribuidor más cercano, o llame o escriba a:



Dirección comercial:
Av. Blasco Ibáñez, 114-116.
46022-Valencia.
Tel. (96) 372 88 89.
Télex 62220

Delegación en Cataluña:
C/ Muntaner, 60, 4, 1.
08011-Barcelona.
Tel. (93) 323 32 19.

ESTOS SON NUESTROS MODELOS:

Modelo	Velocidad	Columnas	Tipos de letra	Interface	P.V.P.
GP-50	40 cps	46	2	A-Paralelo AS-Serial S-Spectrum	A-25.900 AS-29.900 S-28.900
GP-500	50 cps	80	2	A-Paralelo AS-Serial	A-47.900 AS-49.900
GP-550	86 cps	80-136	18	A-Paralelo	A-59.900
GP-700	50 cps	80-106	3	A-Paralelo	A-89.900
BP-5200	200 cps	136-272	18	Paralelo y serial	199.000
BP-5420	420 cps	136-272	18	Paraleo y serial I-IBM PC	299.000 I-299.000

Disponemos de interfaces opcionales para todos los modelos: IBM PC, COMMODORE 64, ZX SPECTRUM, ATARI, DRAGON 64, SHRAP MZ 700, SPECTRAVIDEO, NEW BRAIN, APPLE, ETC...